



**FACULTY  
OF MATHEMATICS  
AND PHYSICS**  
Charles University

**BACHELOR THESIS**

Olga Pribytkova

**Discourse relations of the Prague  
Discourse Treebank in Universal  
Dependencies**

Institute of Formal and Applied Linguistics

Supervisor of the bachelor thesis: RNDr. Jiří Mírovský, Ph.D.

Advisor of the bachelor thesis: Mgr. Lucie Poláková, Ph.D.

Study programme: Computer Science with  
specialisation in Artificial  
Intelligence

Prague 2025

I declare that I carried out this bachelor thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date .....

Author's signature

Dedication. To my family – my parents and husband – who supported me during my education, and to my supervisor and advisor, without whom this work wouldn't be possible.

Title: Discourse relations of the Prague Discourse Treebank in Universal Dependencies

Author: Olga Pribytkova

Institute: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Jiří Mírovský, Ph.D., Institute of Formal and Applied Linguistics

Advisor: Mgr. Lucie Poláková, Ph.D., Institute of Formal and Applied Linguistics

Abstract: This thesis introduces a novel approach for integrating discourse relation annotations from the Prague Discourse Treebank into the Universal Dependencies framework. By aligning PDiT annotations transformed into a PDTB-like format with UD's syntactic data generated by UDPipe, we have created a unified representation that links discourse relations to their corresponding syntactic structures. We then conducted machine learning experiments in discourse type classification using this new format, evaluating feature contribution and performance, highlighting the benefits and challenges of the proposed approach and paving the way for further advancements in computational discourse analysis.

Keywords: discourse relations, Universal Dependencies, machine learning

Název práce: Diskurzivní vztahy Pražského diskurzivního korpusu v Universal Dependencies

Autor: Olga Pribytkova

Ústav: Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Jiří Mírovský, Ph.D., Ústav formální a aplikované lingvistiky

Konzultantka bakalářské práce: Mgr. Lucie Poláková, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Tato diplomová práce navrhuje nový přístup k integraci anotace diskurzivních vztahů Prague Discourse Treebanku do rámce Universal Dependencies. Spojením anotací PDiT transformovaných do formátu podobného PDTB se syntaktickými daty UD generovanými pomocí UDPipe jsme vytvořili jednotnou reprezentaci, která spojuje diskurzivní vztahy s jejich odpovídajícími syntaktickými strukturami. Následně jsme provedli experimenty strojového učení v klasifikaci diskurzivních typů s využitím tohoto nového formátu, při nichž jsme hodnotili příspěvky jednotlivých rysů a výkon modelů, zdůraznili jsme výhody a výzvy navrhovaného přístupu a položili základy pro další pokroky v automatické analýze diskurzu.

Klíčová slova: diskurzivní vztahy, Universal Dependencies, strojové učení

# Introduction

This thesis investigates the integration of discourse relation annotations with surface syntax parsing by experimenting with automatic discourse relation type classification for known arguments and connectives. In particular, in the first part of this work, we will focus on representing discourse relations, which were originally annotated in the Prague Discourse Treebank (PDiT), within the Universal Dependencies (UD) framework. The goal is to take advantage of the syntactic information provided by UD to create a unified representation that can support both linguistic analysis and subsequent machine learning experiments.

The motivation for this research comes from the observation that existing discourse relation annotations (such as those in the PDiT) are often designed independently from automatic syntactic parsing frameworks. By bringing PDiT represented in the format of Penn Discourse Treebank (PDTB) and the UD annotation schemes together, this work aims to enhance the representation of discourse relations. The proposed format will align discourse-level information (such as connective and argument spans) and syntactic structures, which might lead to improved performance in tasks like automatic classification of discourse relation types, which is the focus of the second part of our study.

The experimental part of this thesis is dedicated to utilising the proposed format for automated discourse relations classification through a series of machine learning experiments. The experiments are designed to classify discourse relation types using features extracted from the unified annotations.

In the first part of this work, we will start by providing background on both PDiT and UD. We will outline the key aspects of each and review a related approach to discourse annotation in UD. We will then describe the proposed representation of discourse relations in UD, including the mapping strategy and the format used to embed discourse information into syntactically parsed data. Finally, we will discuss data processing and its results.

In the second part of this work, we will first define two baselines to provide a performance reference. Then advanced machine learning techniques will be explored. For each model, various features derived from the proposed format will be systematically evaluated. Additionally, an error analysis will be conducted, in which we will attempt to uncover common patterns of misclassification and to understand whether there are limitations caused by ambiguities in discourse annotations.

To conclude this thesis, we will discuss the results of both the experiments and the misclassification analysis, along with summarising the overall work done in the study.

**Writing assistance** We used Grammarly<sup>1</sup> and Writerfull<sup>2</sup> for grammar and style corrections, as well as improving text coherence.

---

<sup>1</sup>[www.grammarly.com](http://www.grammarly.com)

<sup>2</sup>[www.writefull.com](http://www.writefull.com)

## Part I

# Representation and Integration of Discourse Relations in Universal Dependencies

# 1 Background and Related Work

## 1.1 Prague Discourse Treebank in PDTB

The Prague Discourse Treebank (PDiT)[1] represents a resource in the study of discourse relations within natural language processing. It provides annotations that go beyond simple sentence-level analysis by incorporating detailed information about discourse relations, connectives, and the spans of their associated arguments.

At its core, PDiT is designed to annotate discourse relations in a way that reflects both the surface-level markers (such as explicit connectives) and the deeper relationships between segments of the text. In addition to providing discourse connectives and argument spans, PDiT includes several layers of linguistic information, such as morphological, surface syntax and deep syntax (tectogrammatical) layers[2]. Tectogrammatical trees provide a deeper syntactic representation of the text by capturing complex predicate-argument structures and showing syntactic-semantic relationships that are not immediately visible in the surface syntax.

The development of PDiT has a long history and exists within a broader research context that includes other discourse annotation frameworks, such as the Penn Discourse Treebank (PDTB)[3]. Although both work with discourse relations, PDiT is distinguished by its focus on the specific linguistic characteristics of the Czech language.

However, the version of PDiT that is being used in this thesis<sup>1</sup> has been transformed into a format similar to PDTB (from here referred to as PDiT-PDTB). This version has many of the additional layers removed, including the tectogrammatical trees, to focus on the core discourse relations, connective and argument spans mapped on plain text. This simplified, PDTB-like format is more aligned with our needs and is easier to integrate with syntactic annotations in the Universal Dependencies framework[4], which will be described in the next section.

Key features of the format that we will utilise in this work include the following:

- **Discourse Relation Typology:** Each relation in PDiT-PDTB is annotated with information indicating the relation type (Explicit, AltLex, AltLexC) and its specific sense class from PDTB (Appendix A.1). It also includes original discourse types in the Prague taxonomy (Appendix A.2).<sup>2</sup>
- **Connectives and Argument Spans:** PDiT-PDTB identifies the discourse connectives and the arguments for each relation by providing character spans corresponding to their positions in the raw text. This way, discourse information is maintained separately from the raw text, yet remains aligned with it through defined character ranges.<sup>3</sup>

---

<sup>1</sup><https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-4875>

<sup>2</sup>It is important to note that discourse types in Prague taxonomy contain no information on argument semantics, such as which specific argument carries a particular role in the relation. This information is not expressed by the original discourse type label after conversion from PDiT to PDiT-PDTB, but is still accessible in the class sense.

<sup>3</sup>The actual words for both the connectives and arguments are given as well, but we will not be using them.

The relation types, as defined by the Penn Discourse Treebank Version 3.0[3], are:

- **Explicit:** These relations are marked by explicit connectives, which are basic conjunctions, adverbs, or particles. Such connectives are part of a relatively closed set of expressions that signal discourse relations directly between clauses or sentences.
- **AltLex:** These relations involve alternative lexicalizations, meaning they are expressed by a variety of linguistic forms or phrases that carry a connective meaning but are not as fixed as explicit relations. AltLex relations span across different sentences (i.e., inter-sentential).
- **AltLexC:** Similar to AltLex, AltLexC also involves alternative lexicalizations, but these relations occur within the same sentence (i.e., intra-sentential).
- **Implicit:** An additional type that is not present in PDiT-PDTB, but is important to mention. Implicit relations do not have a specific connective. They are inferred from the context and the meaning of the connected clauses or sentences.

PDiT-PDTB allows for a nuanced analysis of the discourse by breaking down the discourse structure into detailed components, which later can be used in combination with other formats built on the same collection of raw texts.

For this thesis, PDiT-PDTB serves as a resource that contains detailed discourse annotations. The work we present in this part builds on previous research by proposing a representation of these annotations within the UD framework. By doing so, we aim to combine the strengths of both PDiT-PDTB and UD, so that there is a more integrated approach to discourse analysis that can be used in automatic classification experiments.

In the following example, we will show how different attributes of a discourse relation are represented in PDiT-PDTB. Different fields are separated by vertical bars (|). For example, the first field is the relation type, followed by spans mapped on plain text corresponding to the connective, while the last field is the document type.

The sentence itself is an example of a discourse relation with a discourse type **condition**, which is expressed by the connective *jestli tak* (translation: "if then"), which serves as a link between the arguments *je frajer* (translation: "he is a tough guy") and *tam pojede* (translation: "he will go there").

#### **Example:**

- **Raw text:** *Jestli je frajer, tak tam pojede.*
- **Translation:** *If he is a tough guy, then he will go there.*
- **PDiT-PDTB:**

```

Explicit|1917..1923;1935..1938|||||jestli tak|Contingency.
Condition.Arg2-as-cond
|||||1939..1949|||||1924..1933|||||||1917..1923;1935..1938|||
t-ln94202-128-p6s5w2|condition|Jestli tak|||tam pojede||je
frajer|||comment

```

## 1.2 Universal Dependencies (UD) framework

The Universal Dependencies framework[4] is a system for syntactic annotation that provides a unified approach to analysing sentence structures across different languages. It also allows for language-specific extensions. By defining a standardised set of part-of-speech tags, morphological features, and syntactic dependencies, UD ensures consistent annotation of grammar[5].

The framework is based on dependency syntax where words in a sentence are connected by directed relations that describe their syntactic functions. The relations are head-dependent, meaning that every word (except for the root) depends on another word. This data representation has multiple advantages for discourse analysis. For example, the dependency tree structure enables the mapping of discourse relations (using connective and argument spans) to syntactic units, so there is consistency between syntactic and discourse-level information.

Additionally, the consistent annotation scheme of UD enables comparative studies and multilingual applications by providing a common ground for analysing syntactic structures. Furthermore, there are state-of-the-art syntactic parsers available for many languages that produce UD-compliant data.

To generate UD-compliant syntactic annotations from raw text, we decided to use UDPipe[6], an open-source pipeline for tokenization, tagging, lemmatization and dependency parsing, designed specifically to work within the UD framework. While manually created UD annotations on the collection of texts that we use do exist, they lack token span information, which we need for mapping with the connective and arguments spans from PDiT-PDTB. An additional problem with the manually annotated UD is that the raw text collection used for this annotation is not identical to the one used in PDiT-PDTB in terms of formatting, so the spans of tokens may differ.

UDPipe takes the raw text as input and produces a parsed version of the text, which includes TokenRange values that specify the character offsets for each token.

We argue that, firstly, automated parsing ensures that the same set of rules is applied uniformly across the entire dataset. Secondly, since we are using the same raw text collection for creating the UD as was created during the PDiT-PDTB transformation, the alignment of spans of tokens will be identical. Finally, automated parsing also provides an opportunity for further work with other collections of texts that do not have existing manual UD annotations. Overall, we decided that the use of UDPipe is optimal.

Note that for UD annotations we used `czech-pdt-ud-2.12-230717` model, but now there is a newer model available.<sup>4</sup>

<sup>4</sup>The UDPipe model was trained on the data we are using it to parse, therefore the results of parsing may be better than those on the unseen data

For this thesis, the UD framework is used as a foundation with syntactic information in which we integrate discourse annotations from PDiT-PDTB. Such a combination creates a suitable format for the subsequent experiments in automatic discourse classification. This way, the discourse annotations are easily accessible and their syntactic information can be used as features.

**Example:**

- **Raw text:** *Jestli je frajer, tak tam pojede.*
- **Translation:** *If he is a tough guy, then he will go there.*
- **UD annotation:**

```
# sent_id = 18
# text = Jestli je frajer, tak tam pojede.
1 Jestli jestli SCONJ J,----- _ 2 mark _ TokenRange
  =1917:1923
2 je být AUX VB-S---3P-AAI-- Aspect=Imp|Mood=Ind|Number=Sing|
  Person=3|Polarity=Pos|Tense=Pres|VerbForm=Fin|Voice=Act 7
  advcl _ TokenRange=1924:1926
3 frajer frajer NOUN NNMS1-----A---- Animacy=Anim|Case=Nom|
  Gender=Masc|Number=Sing|Polarity=Pos 2 nsubj _ SpaceAfter=
  No|TokenRange=1927:1933
4 , , PUNCT Z:----- _ 2 punct _ TokenRange=1933:1934
5 tak tak ADV Db----- PronType=Dem 7 advmod _
  TokenRange=1935:1938
6 tam tam ADV Db----- PronType=Dem 7 advmod _
  TokenRange=1939:1942
7 pojede jet VERB VB-S---3F-AAI-- Aspect=Imp|Mood=Ind|Number=
  Sing|Person=3|Polarity=Pos|Tense=Fut|VerbForm=Fin|Voice=Act
  0 root _ SpaceAfter=No|TokenRange=1943:1949
8 . . PUNCT Z:----- _ 7 punct _ SpacesAfter=\n\n|
  TokenRange=1949:1950
```

### 1.3 Existing Approaches to Discourse Annotation in UD

One notable example of a multilayer approach to discourse annotation in UD is the Georgetown University Multilayer (GUM) corpus[7]. In the GUM corpus, the whole text is segmented into elementary discourse units (EDUs) identified through Rhetorical Structure Theory (RST), and since the segmentation is continuous, only the starts of EDUs are marked. Discourse annotations are embedded directly into the attributes of specific tokens at the start of such units. This method encodes several layers of information within a single attribute.

**Example:**

- **Raw text:** *Previous Research*
- **GUM annotation:**

```

# newpar
# newpar_block = head (1 s)
# sent_id = GUM_academic_exposure-10
# s_prominence = 2
# s_type = frag
# transition = zero
# text = Previous Research
1 Previous    previous    ADJ JJ    Degree=Pos 2 amod 2:amod
   Discourse=organization-heading:37->74:9:grf-ly-|Entity=(2-
   abstract-giv:inact-cf1-2-coref
2 Research    research    NOUN NN    Number=Sing 0 root 0:root
   Entity=2)

```

While this rich annotation is excellent for reconstructing RST trees and for in-depth analyses of discourse structure at the EDU level[8], we think it is not suitable for the type of experiments we are conducting in this thesis.

The second part of our work focuses on the automatic classification of discourse relation types using syntactically parsed data in the UD framework. For this, we need a clear and easily extractable representation of discourse relations that aligns with the sentence-level structure of the UD. The complexity of the token-level annotations in the GUM approach adds an unnecessary layer of processing when our primary interest is to map discourse relations directly to the UD trees.

We conclude that while the GUM corpus offers a detailed and comprehensive multilayer annotation that may be effective for certain types of discourse analysis, it is complex, and its reliance on EDU-level segmentation makes it less suitable for our thesis.

# 2 Proposed Representation of Discourse Relations in UD

## 2.1 Possible Approaches

The integration of discourse relation information from PDiT-PDTB to UD is an important part of our study. We considered different approaches to this integration, each with its own advantages and disadvantages. Here, we discuss three main approaches:

### **Distributing Relation Information Across Corresponding Tokens**

This strategy would involve embedding segments of the discourse relation information into the tokens that correspond to various parts of the relation, depending on whether they belong to the connective or the argument spans.

Distributing information in this way would allow each token to carry a part of the discourse relation, which might be useful for detailed token-level analysis. Then each token in the relation can be annotated with its specific role in the relation. However, it also introduces fragmentation and unnecessary complexity. This way, the information is scattered across multiple tokens, which would make it difficult to reassemble the full relation during analysis. The annotation schema also becomes more complicated, and the preprocessing overhead required to collect and interpret the relation data from several token attributes also increases.

### **Embedding Full Relation Information into Identified Connectives**

In this method, the complete information about a discourse relation (relation type, discourse type, connective and argument spans) would be embedded directly into the UD annotation of the tokens that are identified as connectives. The relation details are then directly linked to the connective tokens, making it easy to identify which discourse relation a connective triggers. Nevertheless, this method is still not efficient enough – connectives may consist of multiple words across multiple sentences, which would lead to redundancy. Additionally, this approach would not work for datasets with Implicit relations in future works (since Implicit relations do not have any connectives in the text).

### **Embedding Relation Information as a Comment in the Sentence Header**

Finally, in this approach, the entire discourse relation information would be consolidated and added as a comment in the header of the sentence where the connective is most directly identified. The logic behind this placement is to attach the comment to the first sentence in which the connective appears. This would ensure that the annotation is localized to the sentence where the connective is most prominently identified, while still having access to any related token matches from other sentences within the same comment.

Since we are working with the dataset where there are currently no Implicit relations, we decided that the placement relative to the connective would be preferable. However, for future integration of Implicit relations, the placement

can be easily changed to the sentence where, for example, the first argument is most prominently identified.

Overall, all relevant discourse information for a sentence would be available in a single clearly marked location, which would simplify the extraction of features for further processing and analysis. With this approach, we would also avoid duplicating information across multiple tokens and prevent any conflicts between annotations.

Note that placing the discourse relation comment in the header of the relevant sentence, rather than grouping all comments at the beginning of the document, aligns with the sentence-level logic of UD. This approach combines annotations by embedding discourse relation information directly into the syntactic context, which facilitates both automated processing and human comprehension. Additionally, it is natural for the reader to associate the discourse information with the corresponding sentence this way, so the overall structure is more intuitive and human-readable. This approach is also more robust in case a document is split into several different ones.

There is a potential risk that some token-specific details might be less immediately accessible. However, this is mitigated by the fact that all necessary details are included in the consolidated comment, and a mapping system that we describe in the later section will allow for easy access to the information required.

## 2.2 Format Description

We decided to adopt the third approach: embedding discourse relation information as a comment in the sentence header. The chosen format for the relation comment consolidates all necessary details of a discourse relation in a structured and machine-readable manner. Below is the template of the format:

```
# discourse | index | relation type | class sense | type from PT |  
  sent_id: connective span | sent_id: arg1 span | sent_id: arg2  
  span
```

This comment is composed of several fields, separated by vertical bars (|). Each field has a specific purpose:

1. **Index of the Discourse Relation:** The unique id the discourse relation has within the document. By enumerating the relations, we get straightforward cross-referencing. Each relation can be distinctly tracked throughout the data.
2. **Relation Type:** The relation type (Explicit, AltLex or AltLexC), which were discussed in the previous chapter. Note again that there are no Implicit relations in the dataset we will be working with. However, the proposed format is suitable to support the future inclusion of Implicit relations.
3. **Class Sense:** The fine-grained class sense of discourse relation according to the classification (Appendix A.1).
4. **Prague Taxonomy Type:** The original discourse type in the Prague taxonomy (Appendix A.2), which will later be used as classes for the classification task in our experiments.

5. **Connective Span:** The span of the discourse connective. The span is in the format `sent_id: token_range`, where `sent_id` identifies the sentence containing the connective, and `token_range` lists the specific token (or tokens) which compose the connective. This is used for efficient mapping of the connective information, which is provided by PDiT-PDTB, to the specific token’s information provided by UD.
6. **Arg1 Span:** The span of the first argument of the discourse relation.<sup>1</sup>. The span is in the same `sent_id: token_range` format. This makes it clear where the first argument is located within the text, so discourse relation boundaries can be accurately aligned with UD.
7. **Arg2 Span:** The span of the second argument, similar to Arg1 Span.

The format we propose is robust enough to handle situations where connectives and arguments are located in different sentences or span multiple sentences. For example:

- **Different Sentences:**

**Example:**

```
# discourse | 8 | Explicit | Expansion.Conjunction |
  conjunction | 9: 5 | 8: 1..21 | 9: 1..4 6..17
```

In this case, the connective and one of the arguments are in sentence 9, while the other argument is in sentence 8. By explicitly stating the sentence identifier with each token span, the format clearly points to the exact location of each component despite sentence boundaries.

- **Multiple Sentences for a Single Component:**

**Example:**

```
# discourse | 6 | Explicit | Expansion.Substitution.Arg2-as-
  subst | correction | 5: 8 ; 6: 2 | 5: 1..8 | 6: 1 3..4
```

When a discourse component (e.g. the connective) spans multiple sentences, each span is separated by a semicolon (;) and is labelled with a corresponding sentence id. This ensures that even cross-sentence relations are presented in a clear structured manner.

To summarize, the proposed format provides detailed discourse relation information in a centralized, sentence-level comment that is fully aligned with the UD annotations. This format is optimized for automatic discourse relation classification. The annotation is precise, and we will not have a problem with feature extraction in the subsequent machine learning experiments.

The following example showcases how our processing would modify the sentence we looked at in the previous chapter.

---

<sup>1</sup>By PDTB convention, Arg1 is the left argument in coordinated structures or inter-sentential relations, or the governing argument in subordinating structures.

### Example:

- **Raw text:** *Jestli je frajer, tak tam pojede.*
- **Translation:** *If he is a tough guy, then he will go there.*
- **Processing output:**

```
# sent_id = 18
# text = Jestli je frajer, tak tam pojede.
# discourse | 13 | Explicit | Contingency.Condition.Arg2-as-
  cond | condition | 18: 1 5 | 18: 6..7 | 18: 2..3
1 Jestli jestli SCONJ J,----- _ 2 mark _ TokenRange
  =1917:1923
2 je býť AUX VB-S---3P-AAI-- Aspect=Imp|Mood=Ind|Number=Sing|
  Person=3|Polarity=Pos|Tense=Pres|VerbForm=Fin|Voice=Act 7
  advcl _ TokenRange=1924:1926
3 frajer frajer NOUN NNMS1-----A---- Animacy=Anim|Case=Nom|
  Gender=Masc|Number=Sing|Polarity=Pos 2 nsubj _ SpaceAfter=
  No|TokenRange=1927:1933
4 , , PUNCT Z:----- _ 2 punct _ TokenRange=1933:1934
5 tak tak ADV Db----- PronType=Dem 7 advmod _
  TokenRange=1935:1938
6 tam tam ADV Db----- PronType=Dem 7 advmod _
  TokenRange=1939:1942
7 pojede jet VERB VB-S---3F-AAI-- Aspect=Imp|Mood=Ind|Number=
  Sing|Person=3|Polarity=Pos|Tense=Fut|VerbForm=Fin|Voice=Act
  0 root _ SpaceAfter=No|TokenRange=1943:1949
8 . . PUNCT Z:----- _ 7 punct _ SpacesAfter=\n\n|
  TokenRange=1949:1950
```

## 2.3 Processing Challenges

During the data processing, we faced several challenges. The following issues have been particularly notable.

**Mismatch in Spans** There exist cases when the discourse connective identified in the PDiT-PDTB data does not neatly align with token boundaries in UD. In particular, that happens when the connective is embedded as part of a word, leading to mismatches in the defined character spans:

- **Connective identified in PDiT-PDTB:**

```
1729..1731 | .. | ne-
```

- **Corresponding token in UD:**

```
neobjevila | .. |TokenRange=1729:1739
```

As can be seen in the example, this misalignment complicates the matching process since the token's TokenRange in UD only partially covers the connective specified in PDiT-PDTB. We solved it by comparing the start and end positions of the connective from PDiT-PDTB with the corresponding UD token's range, to ensure that the PDTB span falls within the UD token's boundaries.

Our solution allows for partial overlaps while excluding cases where the UD token extends beyond the PDiT-PDTB span.

**Multi-span Tokens** Consider the following example from the UD data:

```
18-19 aby _ _ _ _ _ _ TokenRange=141:144
18 aby aby SCONJ J,----- _ 20 mark _ _
19 by býť AUX Vc----- Mood=Cnd|Person=3|VerbForm=Fin 20
aux _ _
```

In this case, the summary line (first line) contains a `TokenRange` attribute that represents the span for the multi-span token *aby* (translation: *"in order to"*). However, this line does not include any syntactic information, while the following two lines, which provide the full syntactic annotations for the split components of the token, do not have the `TokenRange` attribute. A part of the processing of the data included removing the summary line from the output format while preserving the `TokenRange` attribute in the split components.

## 2.4 Processing Results

Overall, to build a new dataset, we processed the PDiT-PDTB dataset (which is organized into 10 directories) together with the UDPipe-generated UD annotations from the raw text, and combined them into the proposed format according to the rules introduced in this chapter.

The resulting dataset is the basis for the experiments described in the following chapters, where we will evaluate automatic discourse relation classification methods.

## **Part II**

# **Machine Learning Experiments for Discourse Relation Classification**

### 3 Overview of Experiments

This part of our study presents the machine learning experiments conducted to make use of our proposed format. With the first part of the thesis establishing a new representation of discourse relations, we now proceed to leverage this representation for automatic discourse relation classification.

Our experiments are structured to answer the following questions:

- How effective are various machine learning models in classifying discourse relations, using features derived from the proposed format?
- Which aspects of the representation contribute most to the performance?
- What are the challenges that lead to misclassifications?

We designed a series of experiments, which include multiple baselines and advanced machine learning techniques to address these questions.

The experimental setup uses the created dataset for a comparative analysis of different models in Feature Exploration chapter, which include Feed-Forward Neural Network (FFNN), Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN). Additionally, a division into four main classes, model and baseline ensemble, majority voting and a transformer-based model are introduced to further investigate the influence of the proposed format and compare their classification results to the best model we choose by the end of feature exploration.

For each sample, we have the connective span and the spans for Arg1 and Arg2, which are given in the discourse relation comment. The discourse type (Appendix A.2), which is also located in the discourse relation comment, is the gold annotation.

We decided to classify on original discourse types and not on class senses (Appendix A.1) due to the fact that the discourse types are the original manual annotation of the data in PDiT. Transformation to PDTB class senses was done partially automatically and was not checked manually everywhere, which makes discourse types more reliable[9].

All the arguments to be used as features for model training are extracted from the dataset: we first locate tokens corresponding to the provided spans and then extract information associated with them. The goal is to assign the discourse type to the relation.

In the following chapters, we will first introduce our baseline experiments and then detail the configuration, features, and results of each advanced model. We will also introduce some additional experiments. Afterwards, we will perform an error analysis to identify patterns of misclassification. Finally, we will discuss the challenges that remain unresolved. By conducting these experiments, we aim to explore the performance of various machine learning approaches and to better understand the effectiveness and limitations of the format.

A quick overview of the implementation with instructions on how to recreate some of the experiments is available in **User Guide A.5 in Appendices**.

## 4 Baseline Experiments

In this chapter, we propose baseline approaches that serve as reference points for subsequent advanced machine learning models. The techniques we use here should provide insight into the performance of simpler methods of data classification and establish a strong baseline that our more advanced models will need to outperform (if possible).

As mentioned previously, the dataset is organized into 10 directories:

- 1-8 – training set
- 9 – development set
- 10 – evaluation set

The training set contains 2533 documents, which are composed of 37725 sentences and contain 16763 discourse relations.

The development set contains 316 documents, which are composed of 5126 sentences and contain 2303 discourse relations.

The evaluation set contains 316 documents, which are composed of 5319 sentences and contain 2513 discourse relations.

We use standard evaluation techniques to assess the model performance from scikit-learn `sklearn.metrics`[10], such as:

1. **Precision:** The proportion of positive predictions that are actually correct (how many of the instances labelled as positive are actually positive).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.1}$$

2. **Recall:** The proportion of actual positive instances that are correctly identified (how many of the actual positives were identified).

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.2}$$

3. **F1 Score:** The harmonic mean of Precision and Recall.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

4. **Accuracy:** The proportion of all correct predictions out of all predictions.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \quad (4.4)$$

5. **Macro-averaged F1 Score:** The arithmetic mean of F1 scores for each class. It is calculated independently for each class, then averaged.

$$\text{Macro-avg } F_1 = \frac{1}{n} \sum_{i=1}^n F_1 \text{ Score of class } i \quad (4.5)$$

6. **Weighted-averaged F1 Score:** The average of F1 scores for each class, weighted by the number of true instances for each class (classes with more samples are given more importance, so compared to Macro-avg, the class imbalance is addressed).

$$\text{Weighted-avg } F_1 = \sum_{i=1}^n \frac{\text{True instances of class } i}{\text{Total instances}} F_1 \text{ Score of class } i \quad (4.6)$$

For all metrics:

- **True Positives:** Correctly predicted positive instances.
- **True Negatives:** Correctly predicted negative instances.
- **False Positives:** Instances incorrectly predicted as positive.
- **False Negatives:** Instances incorrectly predicted as negative.

Additionally, in all subsequent classification reports, the **Support** column refers to the number of actual occurrences of each class in the dataset. It shows how many instances of each class are present in the gold annotation. Note that the classification reports report on the results of the evaluation on the development set.

## 4.1 Globally Most Frequent Sense

The first baseline is simple but still informative: it predicts the same discourse relation label for every test instance – the one that appears most frequently in the training data. This approach determines the dominant discourse sense from the training set, then assigns this label to all test cases. It is a very naive method and should be outperformed by any other approach.

Class	Precision	Recall	F1-score	Support
conjunction	0.35	1.00	0.52	816
<b>Accuracy</b>			0.35	2303
<b>Macro avg</b>	0.97	0.05	0.02	2303
<b>Weighted avg</b>	0.77	0.35	0.19	2303

**Table 4.1** Global Most Frequent Type Classification Report on Evaluation Directory

This baseline (Table 4.1) yields an overall accuracy of 0.35. The results show that the majority of instances in the dataset belong to the class `conjunction`, which is the only class with a non-zero recall. In the classification report, we show only `conjunction` since all other classes have precision 1.00 and recall 0.00. This approach does not capture any of the variability or nuances present in the dataset by design.

## 4.2 Most Frequent Type for Each Connective

The second baseline is an improvement from the global prediction strategy because it incorporates connective-specific information. Instead of assigning the same label to every instance, this baseline determines the most frequent discourse type for each individual connective based on the training data. When a test instance contains a connective that was observed during training, the baseline will predict the corresponding most frequent label for this connective. If the baseline encounters a connective not seen in the training set, it will fall back on the overall most common discourse type, which is `conjunction` (as determined by the previous baseline).

Different connectives often signal different types of discourse relations, and this approach takes advantage of that. Some variability for discourse cues can be captured by tailoring predictions to specific connectives. At the same time, when an unseen connective is encountered, the fallback to the overall most common type is an effective enough solution compared to a random guess.

Class	Precision	Recall	F1-score	Support
concession	0.91	0.77	0.83	87
condition	0.78	0.84	0.81	138
confrontation	0.61	0.40	0.48	57
conjunction	0.82	0.95	0.88	816
conjunctive alternative	0.75	0.18	0.29	17
correction	0.89	0.30	0.45	53
disjunctive alternative	0.78	0.94	0.85	34
equivalence	0.43	0.23	0.30	13
explication	0.00	0.00	0.00	16
generalization	1.00	0.07	0.13	14
gradation	0.85	0.73	0.78	62
instantiation	1.00	0.86	0.93	22
opposition	0.82	0.88	0.85	343
pragmatic condition	0.00	0.00	0.00	13
pragmatic contrast	0.00	0.00	0.00	3
pragmatic reason-result	1.00	0.00	0.00	4
precedence-succession	0.75	0.82	0.78	118
purpose	0.93	0.93	0.93	45
reason-result	0.89	0.85	0.87	335
restrictive opposition	0.38	0.10	0.15	31
specification	0.58	0.74	0.65	57
synchrony	0.83	0.20	0.32	25
<b>Accuracy</b>			0.81	2303
<b>Macro avg</b>	0.68	0.49	0.51	2303
<b>Weighted avg</b>	0.80	0.81	0.79	2303

**Table 4.2** Most Frequent Type for Connective Classification Report on Evaluation Directory

This baseline (Table 4.2) achieves an overall accuracy of 0.81. This is a substantial improvement over the previous baseline. The performance on high-support classes is high (*conjunction*, *condition*, *opposition*), which means that this connective-specific mapping is particularly effective when the discourse cues are clear and consistent.

This baseline is particularly strong because discourse connectives are often tied to specific types of discourse relations. They often serve as explicit markers that reflect the interpretation of a discourse relation, and such an approach can capture predictive information. Connectives that consistently signal a particular discourse relation allow the baseline to make accurate predictions in many cases, making it a good point of comparison for more advanced methods. Despite the simplicity of the idea, the results are overall strong.

However, the approach also has its limitations. The performance of this baseline for less frequent or more ambiguous discourse relations is very uneven, which is reflected by their low to moderate F1 scores. We can see that while connective alone is powerful in many cases, it is not enough for the relations that either occur rarely or are not strongly signalled by specific connectives. A few classes (*equivalence*, *pragmatic condition*) have zero precision and recall, due to either insufficient training instances or the possibility that the connectives for

these classes are highly variable or ambiguous.

We will explore improving upon these baselines in the following chapters.

# 5 Models

We briefly introduce the model architecture for each of the models discussed in the next chapter. Each model was trained on directories 1-8, development testing and feature exploration were done on directory 9 (development set), and the evaluation on directory 10 (evaluation set) was done once for each given model with the best set of features, which we choose during feature exploration. In the end, we choose the best overall model with the specific feature configuration based on the results on the development set.

Apart from the models listed below, a Random Forest Classifier was initially explored. However, we decided not to include it in the subsequent feature exploration – it performed worse than other models on the basic feature (connective form and length only), and introducing embeddings for string-based argument features would not have been effective. Random Forest models are not well-suited for handling dense, continuous embeddings, as they rely on splitting features based on discrete, categorical values[11]. Incorporating embeddings would result in large, sparse vectors, which in turn would make the model inefficient and prone to overfitting.

## 5.1 Feed-Forward Neural Network

We consider Feed-Forward Neural Network (FFNN) a good fit for discourse relation classification because it can learn from a mix of features by projecting them into a shared non-linear feature space. Despite its relative simplicity, an FFNN is powerful enough to model complex interactions among the provided cues without a heavier computational overhead[12].

In our implementation, for each discourse relation sample, we extract:

- **Connective Tokens:** Each discourse connective form string is tokenized and mapped to integer IDs via a vocabulary. These IDs are passed through an embedding layer and mean-pooled to produce a fixed-size vector summarising the connective’s semantics. Additionally, the length of the connective (the number of tokens composing the connective) is used as a numeric feature.
- **Numeric Argument Features:** For each argument span, we extract enabled numeric cues – argument length, aggregated polarity score, modal verb counts, and an inter/intra-sentential flag. Each yields two numbers (one per argument), forming a small numeric feature vector.
- **String-Based Argument Features:** For features like roots’ lemmas and immediate dependants’ lemmas, we build a global vocabulary of every unique lemma seen in the training set. During training, each string-based feature is tokenized into vocabulary entries, looked up, and then the model learns an embedding vector for each entry.

These features are explored in more detail in the next chapter.

The pooled connective embedding, the numeric feature vector, and the concatenated string-feature embeddings are joined into one feature vector. This vector is

passed through a fully connected layer with Rectified Linear Unit (ReLU) (5.1) activation and dropout for regularization. A final linear layer maps to logits over the set of discourse relation classes, trained with cross-entropy loss.

$$\text{ReLU}(x) = \max(0, x) \tag{5.1}$$

By concatenating these feature sets into a unified vector and passing them through a non-linear hidden layer, the FFNN learns which combinations of cues best predict each discourse relation.

## 5.2 Convolutional Neural Network

Convolutional Neural Network (CNN) was chosen as another model for the experiments because we use short, local patterns as our connective feature – specific words, short word combinations, whose local token patterns signal the relation type. Convolutional filters are good at detecting patterns between them regardless of the exact position, share parameters across the sequence for efficiency, and require fewer parameters than fully sequential models, which makes them both efficient enough and fast for our task[13].

The CNN model learns local n-gram patterns in the connective sequence and integrates argument-level features.

Each connective is tokenized, mapped through an embedding layer, reshaped into a one-channel image and fed into multiple 2D convolutional filters (kernel sizes 3, 5, and 7), which is followed by ReLU and max-pooling. In parallel, argument features are extracted (numeric and string-based), where each unique string token is looked up in a global vocabulary and mean-pooled via a shared embedding. The string-based features are not handled similarly to connectives due to the fact they can vary in length (from zero up to a dozen or more tokens) and they are noisy, so the gain from learning local word-sequence patterns within them would be marginal compared to the cost of adding extra layers.

## 5.3 Recurrent Neural Network

Recurrent Neural Network (RNN) was also chosen for the task. RNNs excel at modelling variable-length sequences and capturing the temporal dependencies among tokens[12]. Since discourse connectives often span over multiple words whose meaning may depend on their exact order, it seemed interesting to try utilising Long Short-Term Memory (LSTM), which is a type of an RNN cell designed to remember information over long sequences[14]. We feed the connective into an LSTM, and if the connective consists of multiple words, the LSTM captures the exact ordering and interaction of the words. So rather than treating the phrase as a bag of tokens, the LSTM’s hidden state at each step integrates what it’s seen so far, and by the final token, it holds a contextualized summary of the entire multi-word connective.

In the RNN model, each connective is tokenized, mapped through an embedding layer, and fed, in order, into a two-layer LSTM. The final hidden state then serves as the context-sensitive summary. Argument-level features are handled as they

were in the CNN and FFNN experiments: numeric cues are concatenated and projected via a small linear layer, string-based features are tokenized against a global vocabulary, embedded, and mean-pooled to vectors. We did not use any sequence model on them since their order is far less informative and they, again, vary widely in length.

For all models, we chose to represent string-based features with embeddings created by `nn.Embedding`[15] based on custom global vocabularies instead of using pre-trained embeddings for the speed of computation.

# 6 Feature Exploration

In this chapter, we focus our attention on the question of how individual linguistic cues which we extract from the proposed format impact the automated classification of discourse relations.

Our goal is to determine which of these features carry genuine discriminative power when used in different classifiers. We will examine each feature’s contribution to precision, recall, and overall F1 performance.

In addition to the features discussed below, we also tried other features, such as part-of-speech tags and dependency-relation labels of the connective, document types, more parts of the arguments (not just roots or immediate dependants, but other dependants as well), previously predicted discourse types, but it did not yield any meaningful results.

The features we are going to analyse were chosen because each of them has a well-motivated linguistic signal, and they are straightforward in terms of extraction and interpretation.

We will explore the impact of each feature on the resulting performance of the given model. Depending on the results, we will decide whether to include the tested feature in the feature set of the consecutive experiments or not.

The first classification report for each model will be given in full. Afterwards, for the following experiments, only notable changes will be shown for individual classes.

All experiments will be evaluated on the development set, and the best version for each model (FFNN, RNN, CNN) will be evaluated on the evaluation set.

## 6.1 Features

**Connective information** The first feature we propose is the form of the connective, along with its length (the amount of tokens composing the connective). We are choosing the form of the connective and not the lemma since the range of words that can be connectives is not very large, and their forms typically do not differ from their lemmas. And even if they do differ (it happens when the connective is composed of multiple words, still a short sequence), due to word combination it is important to preserve the original form. It is a good start and a point of comparison to the baselines we previously established.

We think that even though each of the models will see only the connective form string, they will still outperform a hard lookup of the most frequent label per connective. If the baseline never saw a connective in training (which would be typical for AltLex and AltLexC relations where the connective is composite and highly variable, so it may not be present in the training set), it simply falls back on the overall most common type.

Our models, however, map each word in the connective into a learned vector space. Synonymous or related connectives end up close together in that space, so the classifier can transfer what it learned about one to the other. Even truly unseen multi-word phrases get tokenized into known words (or into a shared <UNK> embedding) allowing for some non-trivial predictions rather than a blind default. That means that the new connectives in the development data (for which

the baseline must back off to its overall most frequent class) can be correctly routed by the neural models. It gives the models a clear edge in accuracy, even though they do not have any extra context.

**Argument Lengths and Roots’ Lemmas** As the second feature, we introduce basic argument features, such as arguments’ lengths (amounts of tokens in each argument) and arguments’ roots’ lemmas. The length of an argument can indicate its complexity. Differences in length between arguments may reflect discourse patterns.

The root of an argument typically captures its core meaning (often the main verb or predicate), and the lemma of the root provides a normalised semantic representation. In this case, it is beneficial because, in contrast to connectives, the amount of different words that can be roots of arguments is far greater, and the normalised version will still be able to capture the meaning without adding too much noise.

**Polarity** The third feature we propose is polarity, which is computed by aggregating positive and negative cues from the arguments’ information. It provides insight into another aspect of the argument. Different discourse relations may be associated with contrasting polarities. For instance, a mismatch in polarity between arguments can be a strong indicator of a contrastive or adversative relation.

**Argument Root’s Immediate Dependants – Lemmas** The fourth feature we introduce is the argument root’s immediate dependants (children). They carry additional contextual and descriptive information about the argument. Their lemmas may contribute to capturing more nuanced details that may help in better characterising the relation between arguments. Similarly to the roots’ lemmas, it makes more sense to use lemmas here instead of forms since the range of words that can be the root’s immediate dependants is even greater.

**Modal Verbs Counts** The fifth feature we will propose is modal verbs counts. Modal verbs indicate modality (possibility, necessity, uncertainty). Their presence and frequency in arguments may signal different relationships between arguments, such as obligations, possibilities, or contrasts in viewpoint. It adds a dimension of modality and may improve the results of classification.

We chose the following Czech modal verbs, which are called ”true modal verbs” by Daneš et al. (1987)[16]:

- *muset* (translation: ”must”/”have to”)
- *moci* (translation: ”can”/”be able to”)
- *mít* (translation: ”have to”/”must”)
- *smět* (translation: ”may”/”be allowed to”)
- *chtít* (translation: ”want”)
- *hodlat* (translation: ”intend to”/”plan to”)

- *umět* (translation: "be able to"/"know how to")

**Inter/Intra-sentential Flags** The final (sixth) feature we propose is inter/intra-sentential flags. This feature, represented by a binary flag, indicates whether the discourse relation spans across different sentences (inter-sentential) or exists within a single sentence (intra-sentential). This structural information is important because it can signal the nature of a discourse relation[17].

## 6.2 Model Experiments – FFNN

Class	Precision	Recall	F1-score	Support
concession	0.85	0.79	0.82	87
condition	0.78	0.87	0.82	138
confrontation	0.68	0.44	0.53	57
conjunction	0.93	0.95	0.94	816
conjunctive alternative	0.75	0.18	0.29	17
correction	0.82	0.58	0.68	53
disjunctive alternative	0.76	0.94	0.84	34
equivalence	0.43	0.23	0.30	13
explication	0.00	0.00	0.00	16
generalization	0.60	0.21	0.32	14
gradation	0.82	0.81	0.81	62
instantiation	1.00	0.91	0.95	22
opposition	0.78	0.91	0.84	343
pragmatic condition	0.00	0.00	0.00	13
pragmatic contrast	0.00	0.00	0.00	3
pragmatic reason-result	0.00	0.00	0.00	4
precedence-succession	0.76	0.92	0.83	118
purpose	0.94	0.98	0.96	45
reason-result	0.88	0.88	0.88	335
restrictive opposition	0.33	0.10	0.15	31
specification	0.57	0.74	0.64	57
synchrony	0.62	0.20	0.30	25
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.60	0.53	0.54	2303
<b>Weighted avg</b>	0.82	0.84	0.83	2303

**Table 6.1** FFNN Classification Report – Experiment 1.1

**Experiment 1.1 – Connective** The FFNN with the connective information only as a feature (Table 6.1) clearly outperforms the second baseline (Table 4.2). All overall metrics (accuracy, macro and weighted avg) rise, and the improvement can be seen specifically for mid-frequency classes such as **confrontation**, **correction**, **generalization** and **precedence-succession**. There is a bump in F1 even for high-support class **conjunction**.

**Experiment 1.2 – Argument Lengths and Roots’ Lemmas** In this experiment, we introduced argument lengths and roots’ lemmas as additional features.

Class	Precision	Recall	F1-score	Support
equivalence	0.62	0.38	0.48	13
synchrony	0.82	0.36	0.50	25
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.62	0.54	0.55	2303
<b>Weighted avg</b>	0.83	0.84	0.83	2303

**Table 6.2** FFNN Classification Report – Experiment 1.2

As can be seen from the classification report (Table 6.2), accuracy remains at 0.84, but macro avg received a small bump. All high-support classes remain strong, and the low-support classes that had 0.00 in all metrics remain unsolved.

However, in this experiment, it is interesting to pay attention to classes like **synchrony** and **equivalence**. Although the difference between precision and recall for both of them was and remains big, meaning that many of the positive cases are being missed, both precision and recall increased, which is reflected in a greater F1 score. We will keep these features for future experiments.

**Experiment 1.3 – Polarities** We are now introducing polarities to the existing set of features.

Class	Precision	Recall	F1-score	Support
equivalence	0.67	0.62	0.64	13
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.62	0.55	0.56	2303
<b>Weighted avg</b>	0.83	0.84	0.83	2303

**Table 6.3** FFNN Classification Report – Experiment 1.3

From the classification report (Table 6.3), we can see that accuracy once again did not improve, and macro avg again got a bit better. The only interesting change for individual classes occurred for **equivalence**, with its F1 score rising due to the rise in recall – with the introduction of polarity, more of the missed cases are now being identified for this class. The polarity will also remain in the future feature sets.

**Experiment 1.4 – Argument Roots’ Immediate Dependants** We now add argument roots’ immediate dependants to the existing features.

Class	Precision	Recall	F1-score	Support
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.62	0.54	0.55	2303
<b>Weighted avg</b>	0.82	0.84	0.82	2303

**Table 6.4** FFNN Classification Report – Experiment 1.4

There are no clear improvements in the metrics (Table 6.4), with both macro and weighted avg dropping slightly due to decreases in most of mid-frequency and low-support classes. This feature did not add anything meaningful to the model and will not be in feature sets for future experiments on FFNN.

**Experiment 1.5 – Modal Verbs Counts** With the addition of modal verbs counts (Table 6.5), we see an improvement in accuracy for the first time, and macro and weighted avg also increase a little. There are slight changes in F1 scores for some mid-frequency classes (some see an increase, some see a decrease), but there are no notable drops nor improvements compared to the **Experiment 1.4** (Table 6.3). This feature will also be preserved for the last experiment.

Class	Precision	Recall	F1-score	Support
<b>Accuracy</b>			0.85	2303
<b>Macro avg</b>	0.63	0.55	0.56	2303
<b>Weighted avg</b>	0.83	0.85	0.83	2303

**Table 6.5** FFNN Classification Report – Experiment 1.5

**Experiment 1.6 – Inter/Intra-sentential Flags** After we introduce this last feature to our set of features (Table 6.6), we see a decrease in both accuracy and macro avg scores. The most drastic drops for the classes occur for **synchrony** and **equivalence**.

Class	Precision	Recall	F1-score	Support
equivalence	0.50	0.31	0.38	13
synchrony	0.54	0.28	0.37	25
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.59	0.54	0.55	2303
<b>Weighted avg</b>	0.82	0.84	0.83	2303

**Table 6.6** FFNN Classification Report – Experiment 1.6

**Conclusion and The Best Model** As can be seen throughout the experiments, there is little impact of particular features on the performance of our FFNN model – high-support classes remain strong with equally high precision and recall, while low-support classes remain unseen, with four of them (**explication**, **pragmatic condition**, **pragmatic contrast** and **pragmatic reason-result**) never improving above 0.00. The most affected classes were **synchrony** and **equivalence**.

Based on the development data, we consider the model from **Experiment 1.5** (Table 6.5) to be the best FFNN model. Its feature set contains: connective form and length, argument lengths, argument roots’ lemmas, polarity, modal verbs counts.

Class	Precision	Recall	F1-score	Support
concession	0.90	0.72	0.80	92
condition	0.74	0.92	0.82	165
confrontation	0.77	0.59	0.67	73
conjunction	0.95	0.94	0.94	904
conjunctive alternative	1.00	0.31	0.47	13
correction	0.96	0.68	0.80	38
disjunctive alternative	0.72	1.00	0.84	26
equivalence	0.20	0.09	0.12	11
explication	0.00	0.00	0.00	11
generalization	0.50	0.33	0.40	15
gradation	0.88	0.77	0.82	64
instantiation	1.00	1.00	1.00	18
opposition	0.79	0.97	0.87	366
pragmatic condition	0.00	0.00	0.00	20
pragmatic contrast	0.00	0.00	0.00	2
pragmatic reason-result	0.00	0.00	0.00	4
precedence-succession	0.79	0.86	0.83	145
purpose	0.93	0.98	0.95	41
reason-result	0.90	0.91	0.90	349
restrictive opposition	0.43	0.09	0.15	34
specification	0.59	0.86	0.70	76
synchrony	0.69	0.20	0.31	46
<b>Accuracy</b>			0.86	2513
<b>Macro avg</b>	0.63	0.55	0.56	2513
<b>Weighted avg</b>	0.84	0.86	0.84	2513

**Table 6.7** Best FFNN Classification Report on Evaluation Set

In the classification report on the evaluation set (Table 6.7), we can see the model performs well on the unseen data. The class `instantiation` even achieves perfect 1.00 for precision, recall and F1 while being a low-support class.

### 6.3 Model Experiments – RNN

**Experiment 2.1 – Connective** The RNN with the connective form and length as features (Table 6.8) also performs better than the second baseline (Table 4.2), although the results are worse compared to FFNN with the same feature (Table 6.1). While the accuracy is the same, the macro avg is noticeably worse. High-support classes like `conjunction` or `reason-result` still get an improvement from the baseline, as well as some mid-frequency classes: `gradation` and `correction` improve. However, other classes struggle, notably both `synchrony`, `restrictive opposition` and `equivalence` are not detected at all.

Class	Precision	Recall	F1-score	Support
concession	0.87	0.78	0.82	87
condition	0.76	0.87	0.81	138
confrontation	0.71	0.44	0.54	57
conjunction	0.93	0.95	0.94	816
conjunctive alternative	0.00	0.00	0.00	17
correction	0.78	0.75	0.77	53
disjunctive alternative	0.67	0.91	0.78	34
equivalence	0.00	0.00	0.00	13
explication	0.00	0.00	0.00	16
generalization	0.33	0.14	0.20	14
gradation	0.78	0.76	0.77	62
instantiation	0.67	0.91	0.77	22
opposition	0.80	0.90	0.85	343
pragmatic condition	0.00	0.00	0.00	13
pragmatic contrast	0.00	0.00	0.00	3
pragmatic reason-result	0.00	0.00	0.00	4
precedence-succession	0.73	0.92	0.82	118
purpose	0.94	0.98	0.96	45
reason-result	0.86	0.90	0.88	335
restrictive opposition	0.00	0.00	0.00	31
specification	0.61	0.74	0.67	57
synchrony	0.00	0.00	0.00	25
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.47	0.50	0.48	2303
<b>Weighted avg</b>	0.80	0.84	0.81	2303

**Table 6.8** RNN Classification Report – Experiment 2.1

**Experiment 2.2 – Argument Lengths and Roots’ Lemmas** With the introduction of basic argument features (Table 6.9), the model starts to struggle noticeably. The accuracy decreases to 0.81, and both weighted and macro avg decrease as well. While there are some improvements – F1 for `correction` and `instantiation` increases while `equivalence`, `pragmatic condition`, `restrictive opposition` and `synchrony` rise from their 0.00 in the previous experiment (Table 6.8), most of the other classes suffer. This feature therefore will not be included in all future experiments’ feature sets.

Class	Precision	Recall	F1-score	Support
correction	0.68	0.68	0.68	53
equivalence	0.25	0.08	0.12	13
instantiation	0.89	0.77	0.83	22
pragmatic condition	0.50	0.08	0.13	13
restrictive opposition	0.75	0.10	0.17	31
synchrony	0.36	0.20	0.26	25
<b>Accuracy</b>			0.81	2303
<b>Macro avg</b>	0.54	0.48	0.48	2303
<b>Weighted avg</b>	0.79	0.81	0.79	2303

**Table 6.9** RNN Classification Report – Experiment 2.2

**Experiment 2.3 – Polarities** With the introduction of polarities (Table 6.10), the overall metrics improve compared to the previous experiment (Table 6.9). And although the accuracy is slightly worse than in the **Experiment 2.1**, the macro avg is better here, as well as performance on some of the low-support classes, such as `equivalence`, `instantiation`, `restrictive opposition` and `synchrony`. Even if F1 score for some well-represented classes drops a little (`concession`, `precedence-succession`), we think it is worth it to keep this feature in our feature set for future experiments.

Class	Precision	Recall	F1-score	Support
concession	0.78	0.79	0.78	87
equivalence	0.50	0.08	0.13	13
instantiation	0.91	0.91	0.91	22
precedence-succession	0.68	0.90	0.78	118
restrictive opposition	0.50	0.10	0.16	31
synchrony	0.75	0.12	0.21	25
<b>Accuracy</b>			0.83	2303
<b>Macro avg</b>	0.56	0.50	0.50	2303
<b>Weighted avg</b>	0.81	0.83	0.81	2303

**Table 6.10** RNN Classification Report – Experiment 2.3

**Experiment 2.4 – Argument Roots’ Immediate Dependants** As we introduce argument roots’ immediate dependants as our next feature (Table 6.11), we once again see a decline in all overall metrics. Most of the mid-frequency classes suffer (`restrictive opposition` even drops to 0.00) with the exception of `confrontation` and `disjunctive alternative`, and notably, `synchrony` improves and `conjunctive alternative` is detected for the first time, although the difference between the precision and recall shows that it misses the positive instances of this class too often. Overall, it is another string-based feature that worsened the performance of the model, and it will not be included in future feature sets.

Class	Precision	Recall	F1-score	Support
confrontation	0.59	0.40	0.48	57
conjunctive alternative	0.50	0.06	0.11	17
disjunctive alternative	0.70	0.88	0.78	34
restrictive opposition	0.00	0.00	0.00	31
synchrony	0.54	0.28	0.37	25
<b>Accuracy</b>			0.82	2303
<b>Macro avg</b>	0.49	0.47	0.47	2303
<b>Weighted avg</b>	0.78	0.82	0.79	2303

**Table 6.11** RNN Classification Report – Experiment 2.4

**Experiment 2.5 – Modal Verbs Counts** The modal verbs counts as a feature for RNN did not improve anything also (Table 6.12). The accuracy is still lower than in **Experiment 2.1** (Table 6.8) and **Experiment 2.3** (Table 6.10), and macro and weighted avg does not improve either. The performance across classes has also decreased in comparison to the previously mentioned experiments without any notable improvements, therefore we will not be using this feature in the next experiment.

Class	Precision	Recall	F1-score	Support
<b>Accuracy</b>			0.83	2303
<b>Macro avg</b>	0.50	0.49	0.48	2303
<b>Weighted avg</b>	0.79	0.83	0.81	2303

**Table 6.12** RNN Classification Report – Experiment 2.5

**Experiment 2.6 – Inter/Intra-sentential Flags** As the last experiment for RNN, we add inter/intra-sentential flags (Table 6.13). The results do not differ much from the **Experiment 2.5** (Table 6.12) – the performance in all overall metrics is still worse than of the better models, and there are no interesting improvements for individual classes.

Class	Precision	Recall	F1-score	Support
<b>Accuracy</b>			0.83	2303
<b>Macro avg</b>	0.52	0.48	0.48	2303
<b>Weighted avg</b>	0.79	0.83	0.80	2303

**Table 6.13** RNN Classification Report – Experiment 2.6

**Conclusion and The Best Model** From the experiments above we can see that RNN did not gain almost anything from additional features, and string-based features that were not the connective form were especially harmful. Polarities, however, introduced a slight improvement in multiple low-support classes with the trade-off in high-support classes, which we found acceptable. Therefore, based on the development data, the model from **Experiment 2.3** (Table 6.10) is chosen as the best RNN model. Its feature set contains: connective form and length, polarity.

Class	Precision	Recall	F1-score	Support
concession	0.90	0.72	0.80	92
condition	0.67	0.88	0.76	165
confrontation	0.81	0.59	0.68	73
conjunction	0.95	0.95	0.95	904
conjunctive alternative	0.00	0.00	0.00	13
correction	0.79	0.61	0.69	38
disjunctive alternative	0.63	1.00	0.78	26
equivalence	0.00	0.00	0.00	11
explication	0.00	0.00	0.00	11
generalization	0.25	0.13	0.17	15
gradation	0.88	0.78	0.83	64
instantiation	1.00	0.94	0.97	18
opposition	0.80	0.96	0.87	366
pragmatic condition	0.00	0.00	0.00	20
pragmatic contrast	0.00	0.00	0.00	2
pragmatic reason-result	0.00	0.00	0.00	4
precedence-succession	0.73	0.90	0.81	145
purpose	0.93	0.98	0.95	41
reason-result	0.90	0.90	0.90	349
restrictive opposition	0.33	0.09	0.14	34
specification	0.66	0.70	0.68	76
synchrony	0.62	0.11	0.19	46
<b>Accuracy</b>			0.85	2513
<b>Macro avg</b>	0.54	0.51	0.51	2513
<b>Weighted avg</b>	0.83	0.85	0.83	2513

**Table 6.14** Best RNN Classification Report on Evaluation Set

The final evaluation on the evaluation set (Table 6.14 shows us that the performance on the unseen data for this model is considerably strong, however, the overall performance is still worse than the one of the best FFNN model.

## 6.4 Model Experiments – CNN

**Experiment 3.1 – Connective** The CNN with the connective information performs best out of both the baseline and other models with the same feature set (Table 6.15), with both the accuracy and macro avg being the highest. However, the results for individual classes do not differ that much, since the high-support classes still perform strongly, while low-support classes that went unnoticed before still remain unnoticed.

Class	Precision	Recall	F1-score	Support
concession	0.87	0.78	0.82	87
condition	0.77	0.87	0.82	138
confrontation	0.74	0.40	0.52	57
conjunction	0.91	0.95	0.93	816
conjunctive alternative	0.60	0.18	0.27	17
correction	0.66	0.85	0.74	53
disjunctive alternative	0.78	0.94	0.85	34
equivalence	0.43	0.23	0.30	13
explication	0.00	0.00	0.00	16
generalization	1.00	0.29	0.44	14
gradation	0.89	0.66	0.76	62
instantiation	1.00	0.91	0.95	22
opposition	0.81	0.90	0.85	343
pragmatic condition	0.00	0.00	0.00	13
pragmatic contrast	0.00	0.00	0.00	3
pragmatic reason-result	0.00	0.00	0.00	4
precedence-succession	0.77	0.94	0.84	118
purpose	0.94	1.00	0.97	45
reason-result	0.87	0.90	0.88	335
restrictive opposition	0.44	0.13	0.20	31
specification	0.60	0.74	0.66	57
synchrony	1.00	0.16	0.28	25
<b>Accuracy</b>			0.85	2303
<b>Macro avg</b>	0.64	0.54	0.55	2303
<b>Weighted avg</b>	0.83	0.85	0.83	2303

**Table 6.15** CNN Classification Report – Experiment 3.1

**Experiment 3.2 – Argument Lengths and Roots’ Lemmas** By introducing these features to the feature set of the CNN (Table 6.16), we did not see any gains. The overall metrics suffered, just like most of the individual classes. The only clear improvement was in class **correction**. Classes **conjunctive alternative** and **synchrony** also saw a spike in F1, but due to the continuous high precision - low recall problem, it is not very meaningful. Therefore, we will not be using these features in further experiments.

Class	Precision	Recall	F1-score	Support
conjunctive alternative	0.80	0.24	0.36	17
correction	0.77	0.83	0.80	53
synchrony	0.86	0.24	0.38	25
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.63	0.52	0.54	2303
<b>Weighted avg</b>	0.82	0.84	0.82	2303

**Table 6.16** CNN Classification Report – Experiment 3.2

**Experiment 3.3 – Polarities** The introduction of polarities (Table 6.17) again did not improve the results. Mid-frequency classes like **concession** and **correction** saw an improvement from the first experiment (Table 6.15), and although F1 scores for **synchrony** and **restrictive opposition** improved, the gap between the precision and recall for them remains large. So while high-support classes continue to show good results, most of the other classes saw a small decline. We will also not be using this feature in all other experiments’ feature sets.

Class	Precision	Recall	F1-score	Support
concession	0.88	0.82	0.85	87
correction	0.86	0.70	0.77	53
restrictive opposition	0.60	0.19	0.29	31
synchrony	0.70	0.28	0.40	25
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.62	0.54	0.55	2303
<b>Weighted avg</b>	0.83	0.84	0.83	2303

**Table 6.17** CNN Classification Report – Experiment 3.3

**Experiment 3.4 – Argument Roots’ Immediate Dependents** The introduction of this feature to the CNN (Table 6.18) did not achieve any changes in the metrics. Accuracy and macro and weighted avg continue to be worse than those of the current best CNN model (Table 6.15), and not a single individual class saw an improvement. We will continue the experiments without this feature.

Class	Precision	Recall	F1-score	Support
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.58	0.52	0.52	2303
<b>Weighted avg</b>	0.82	0.84	0.82	2303

**Table 6.18** CNN Classification Report – Experiment 3.4

**Experiment 3.5 – Modal Verbs Counts** With the introduction of modal verbs counts (Table 6.19), no notable changes happened. High-support classes continue to perform well, low-support classes continue to struggle and nothing major happened to other individual classes, showing that the impact of this feature was minimal. We will proceed to the final experiment without this feature in the feature set.

Class	Precision	Recall	F1-score	Support
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.66	0.52	0.54	2303
<b>Weighted avg</b>	0.83	0.84	0.82	2303

**Table 6.19** CNN Classification Report – Experiment 3.5

**Experiment 3.6 – Inter/Intra-sentential Flags** The final experiment with the inter/intra-sentential flags feature also did not yield any meaningful results

(Table 6.20). There were no improvements in overall metrics and no interesting movements for individual classes.

Class	Precision	Recall	F1-score	Support
<b>Accuracy</b>			0.84	2303
<b>Macro avg</b>	0.62	0.52	0.54	2303
<b>Weighted avg</b>	0.82	0.84	0.82	2303

**Table 6.20** CNN Classification Report – Experiment 3.6

**Conclusion and The Best Model** From this section’s experiments on the development set, we can see that no additional feature was able to improve upon the performance of the connective information for CNN from the **Experiment 3.1** (Table 6.15), which we will consider as the best model for CNN. Its feature set includes only connective form and length.

Class	Precision	Recall	F1-score	Support
concession	0.89	0.70	0.78	92
condition	0.66	0.87	0.75	165
confrontation	0.75	0.58	0.65	73
conjunction	0.95	0.94	0.94	904
conjunctive alternative	0.80	0.31	0.44	13
correction	0.71	0.79	0.75	38
disjunctive alternative	0.71	0.96	0.82	26
equivalence	0.00	0.00	0.00	11
explication	0.00	0.00	0.00	11
generalization	1.00	0.13	0.24	15
gradation	0.91	0.75	0.82	64
instantiation	1.00	1.00	1.00	18
opposition	0.80	0.95	0.87	366
pragmatic condition	0.00	0.00	0.00	20
pragmatic contrast	0.00	0.00	0.00	2
pragmatic reason-result	1.00	0.25	0.40	4
precedence-succession	0.74	0.90	0.82	145
purpose	0.95	0.98	0.96	41
reason-result	0.88	0.92	0.90	349
restrictive opposition	0.30	0.09	0.14	34
specification	0.65	0.72	0.69	76
synchrony	1.00	0.04	0.08	46
<b>Accuracy</b>			0.85	2513
<b>Macro avg</b>	0.67	0.54	0.55	2513
<b>Weighted avg</b>	0.84	0.85	0.83	2513

**Table 6.21** Best CNN Classification Report on Evaluation Set

The performance on the evaluation set is rather strong, with good accuracy and **instantiation** achieving the perfect score and **pragmatic reason-result** rising from 0.00 on development, although the gap between the precision and recall is big.

## 6.5 Conclusion

As was suggested in the beginning, all models outperformed the second baseline even with just connective information.

However, while the proposed features may have appeared useful for improving models' performances in theory, in practice they proved to influence the results very slightly, if at all at times. They are able to influence some low- and mid-frequency classes to some extent.

Certain classes like **synchrony**, **generalization**, **restrictive opposition** continue to suffer from high precision - low recall problem, meaning essentially that it is unlikely the model will believe an actual instance of one of these classes (according to the gold label) to be an instance of the class, and if it does, then the evidence of belonging to the class must have been almost certain.

It most likely happens due to the crossover between the frequent connectives of different classes. For example, for **synchrony**, the word "*když*" (translation: "*when*") is the most frequent connective, but it is also a frequent connective for classes like **concession** or **condition**, which have more support in the dataset. The additional features fail to seriously contribute to differentiating between classes with the same frequent connective. Information about most often misclassified connectives can be found here (Appendix A.4).

In turn, the class **instantiation**, which is of mid- to low-support, achieves either perfect or almost perfect scores across all models. It happens due to the fact that the connective of this class is not very variable – the majority of connectives (whether they are one word or composite) contain the words "*například*", "*příklad*" or "*např*" (translation: "*for example*"), so it is easy for the model with basic connective information or even the baseline to yield good results even if the support for the class is not big.

More analysis on misclassifications will be performed in **Chapter 8**.

To conclude, based on the results of experiments on the development set, we choose the best FFNN model (Table 6.5) to be the best overall model.

# 7 Additional Experiments

In this chapter, some additional experiments are conducted to probe into possible future directions of the research. All the experiments’ classification reports contain evaluations on the evaluation set. They are evaluated straight on the evaluation set due to being shown for strictly illustrative purposes (as we have already chosen our best model), and no selection of best parameters is done.

## 7.1 Four Major Classes

Four major classes are defined in Introducing the Prague Discourse Treebank 3.0 [1]. The full grouping into major classes can be found in Appendix A.3.

Grouping 22 classes (Appendix A.2) into four major classes can be beneficial for comparison in several ways. Firstly, it alleviates data sparsity by pooling classes together, which stabilises model training and reduces the confusion caused by semantic overlap among classes that are closer to each other (e.g. `reason-result` and `pragmatic reason-result`, or `conjunctive alternative` and `disjunctive alternative`). Secondly, by focusing on broader classes, we can more clearly identify which discourse distinctions remain challenging.

First, we run the first baseline (Table 4.1). It shows that the globally most frequent major class is `Expansion`, and this baseline yields an accuracy of 0.42.

Now, let us take a look at the second baseline (Table 4.2) for the four major classes:

Class	Precision	Recall	F1-score	Support
Contingency	0.89	0.88	0.89	590
Contrast	0.94	0.89	0.91	669
Expansion	0.87	0.93	0.90	1063
Temporal	0.80	0.69	0.74	191
<b>Accuracy</b>			0.89	2513
<b>Macro avg</b>	0.88	0.85	0.86	2513
<b>Weighted avg</b>	0.89	0.89	0.89	2513

**Table 7.1** Classification Report for Major Discourse Classes – Baseline

Pooling the classes into major categories helped with improving the baseline’s results, raising the accuracy to 0.89. `Expansion` and `Contrast` dominate overall metrics simply because they are the largest classes, and baseline pushes unseen connectives into whichever class is globally most frequent or locally most frequent for that connective.

When we apply the implementation of the best model (Table 6.5) retrained for this task, the results are uniformly strong (Table 7.2), with now three of the major classes having F1 score greater than 0.90, and all three see an improvement, which accuracy and macro and weighted avg reflect also. The only low point is the scores for the class `Temporal`, which remain worse than for any other class, even with the slight improvement from the baseline. It is clear that the severe underrepresentation in the dataset of the few classes that are included in this

major class, together with the poor performance on one of them (**synchrony**) is the cause of the low scores, which in turn causes a decrease in the overall accuracy and macro-avg F1 as well.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Contingency	0.93	0.90	0.92	590
Contrast	0.95	0.96	0.95	669
Expansion	0.94	0.92	0.93	1063
Temporal	0.72	0.86	0.79	191
<b>Accuracy</b>			0.92	2513
<b>Macro avg</b>	0.89	0.91	0.90	2513
<b>Weighted avg</b>	0.93	0.92	0.92	2513

**Table 7.2** Classification Report for Four Major Classes – Best Model

## 7.2 Baseline and Model Ensemble

This experiment explores a way to combine the best model from **Experiment 5** (Table 6.5) with the Most Frequent Type baseline (Table 4.2) by picking for each connective which predictor to trust. We first evaluate both the baseline and our trained FFNN on the development set, and for each distinct connective we record which method was more accurate. On the evaluation set, whenever that connective appears, we route the example to whichever predictor has been chosen for it previously.

The motivation behind this method is that frequent connectives often have a very stable most-likely relation (the baseline excels there), while more rare connectives benefit from the model’s implementation and additional features. So their combination might boost overall accuracy (by avoiding each method’s weak spots).

Class	Precision	Recall	F1-score	Support
concession	0.90	0.72	0.80	92
condition	0.75	0.91	0.82	165
confrontation	0.77	0.59	0.67	73
conjunction	0.95	0.94	0.95	904
conjunctive alternative	0.80	0.31	0.44	13
correction	0.96	0.68	0.80	38
disjunctive alternative	0.71	0.96	0.82	26
equivalence	0.17	0.09	0.12	11
explication	0.00	0.00	0.00	11
generalization	1.00	0.20	0.33	15
gradation	0.88	0.81	0.85	64
instantiation	1.00	1.00	1.00	18
opposition	0.79	0.97	0.87	366
pragmatic condition	0.00	0.00	0.00	20
pragmatic contrast	0.00	0.00	0.00	2
pragmatic reason-result	0.00	0.00	0.00	4
precedence-succession	0.78	0.87	0.82	145
purpose	0.93	0.95	0.94	41
reason-result	0.90	0.91	0.90	349
restrictive opposition	0.38	0.09	0.14	34
specification	0.59	0.86	0.70	76
synchrony	0.67	0.17	0.28	46
<b>Accuracy</b>			0.86	2513
<b>Macro avg</b>	0.63	0.55	0.56	2513
<b>Weighted avg</b>	0.85	0.86	0.84	2513

**Table 7.3** FFNN and Baseline Ensemble Classification Report on Evaluation Set

As can be seen from the report (Table 7.3), the results are comparable to the best model’s results (Table 6.7), even though in most of the cases (325/405), the baseline was chosen as the predictor. Compared to the best model, performance slightly improved for **gradation**, but also slightly decreased on most of the other classes with the exception of high-support classes.

## 7.3 Voting

As another additional experiment, we built a majority-vote ensemble to see if combining our three best models could yield better predictions. We used the best models from the previous chapter: the FFNN model from **Experiment 5** (Table 6.5), the CNN model from **Experiment 1** (Table 6.15) and the RNN model from **Experiment 3** (Table 6.10).

For each example, we take the three predicted relation labels and choose the one that has been selected most often. If all three models predict a different label, the FFNN prediction is used.

Class	Precision	Recall	F1-score	Support
concession	0.90	0.72	0.80	92
condition	0.67	0.88	0.76	165
confrontation	0.79	0.58	0.67	73
conjunction	0.95	0.95	0.95	904
conjunctive alternative	1.00	0.31	0.47	13
correction	0.97	0.74	0.84	38
disjunctive alternative	0.70	1.00	0.83	26
equivalence	0.00	0.00	0.00	11
explication	0.00	0.00	0.00	11
generalization	0.75	0.20	0.32	15
gradation	0.88	0.80	0.84	64
instantiation	1.00	1.00	1.00	18
opposition	0.79	0.97	0.87	366
pragmatic condition	0.00	0.00	0.00	20
pragmatic contrast	0.00	0.00	0.00	2
pragmatic reason-result	0.00	0.00	0.00	4
precedence-succession	0.75	0.91	0.82	145
purpose	0.93	0.98	0.95	41
reason-result	0.90	0.91	0.90	349
restrictive opposition	0.43	0.09	0.15	34
specification	0.68	0.71	0.69	76
synchrony	1.00	0.09	0.16	46
<b>Accuracy</b>			0.85	2513
<b>Macro avg</b>	0.64	0.54	0.55	2513
<b>Weighted avg</b>	0.85	0.85	0.83	2513

**Table 7.4** Voting Classification Report on Evaluation Set

Comparing this experiment’s results (Table 7.4) to the best model’s results (Table 6.7), we can see that overall accuracy, weighted and macro avg slightly decrease. The only clear improvement happens for the class `correction`, while several other classes see a decline in F1 (`synchrony`, `generalization`, `condition`).

Overall, the voting performed slightly worse than the existing best model.

## 7.4 Transformer

As another additional experiment, we decided to implement a Transformer-based classifier built on top of the pre-trained `ufal/robeczech-base`[18] model. Since it produces contextualized representations, we used connectives and whole arguments as concatenated text to produce contextualized embeddings.

The model delivers the strongest overall performance to date (Table 7.6), achieving 0.88 accuracy, a weighted F1 of 0.86, and a macro-average F1 of 0.60 – each is an improvement over our other best models (Table 7.5). It excels on high-support classes (`condition`, `conjunction`, `precedence-succession`) and raises scores for mid-frequency classes such as `correction`, `generalization`, and `specification`. Notably, it begins to recover some previously weak classes such as `equivalence` and `pragmatic condition`, indicating that the richer contextu-

alized embeddings capture subtler cues. Additionally, the F1 score for the class `synchrony` has risen, and its precision and recall scores became considerably more balanced in comparison to our other models.

However, it still fails on most low-support classes (`conjunctive alternative`, `pragmatic contrast`, `pragmatic reason-result` remain at 0.00), and `restrictive opposition` and `explication` continue to struggle.

Overall, this implementation yields consistent gains across the board, showing that modern large neural network models seem to outperform analytical approaches tried in previous parts of the study, and yet rare and ambiguous classes still cannot be classified.

Model	Accuracy	Macro avg F1	Weighted avg F1
Best FFNN	0.86	0.56	0.84
Best RNN	0.85	0.51	0.83
Best CNN	0.85	0.55	0.83
Transformer	0.88	0.60	0.86

**Table 7.5** Comparison to the Best Models

Class	Precision	Recall	F1-score	Support
<code>concession</code>	0.87	0.78	0.82	92
<code>condition</code>	0.88	0.94	0.91	165
<code>confrontation</code>	0.66	0.60	0.63	73
<code>conjunction</code>	0.95	0.96	0.95	904
<code>conjunctive alternative</code>	0.00	0.00	0.00	13
<code>correction</code>	0.93	0.74	0.82	38
<code>disjunctive alternative</code>	0.63	1.00	0.78	26
<code>equivalence</code>	0.50	0.27	0.35	11
<code>explication</code>	0.33	0.09	0.14	11
<code>generalization</code>	0.56	0.33	0.42	15
<code>gradation</code>	0.89	0.75	0.81	64
<code>instantiation</code>	1.00	1.00	1.00	18
<code>opposition</code>	0.80	0.94	0.87	366
<code>pragmatic condition</code>	0.50	0.30	0.38	20
<code>pragmatic contrast</code>	0.00	0.00	0.00	2
<code>pragmatic reason-result</code>	0.00	0.00	0.00	4
<code>precedence-succession</code>	0.90	0.88	0.89	145
<code>purpose</code>	0.95	0.98	0.96	41
<code>reason-result</code>	0.90	0.93	0.92	349
<code>restrictive opposition</code>	0.50	0.12	0.19	34
<code>specification</code>	0.76	0.80	0.78	76
<code>synchrony</code>	0.69	0.54	0.61	46
<b>Accuracy</b>			0.88	2513
<b>Macro avg</b>	0.65	0.59	0.60	2513
<b>Weighted avg</b>	0.86	0.88	0.86	2513

**Table 7.6** Transformer Classification Report on Evaluation Set



issue (Appendix A.4). For example, **restrictive opposition** and **opposition** labels are very close to each other, and very often the difference between the two is purely semantic, with **restrictive opposition** suffering due to being more niche and harder to classify. And as can be seen from Figure 8.1, it is the most frequent misclassification between any two classes (25 counts). Likewise, the difference between **conjunctive alternative** and **disjunctive alternative** can often be minute, and both interpretations can be possible. Additionally, both **conjunction** and **gradation** act as links between clauses, but they differ in how they connect the ideas. However, even despite the difference, they remain semantically close: in many contexts, the connective that simply joins ideas may also indirectly point at a layering of information.

We will proceed to analyse some examples in detail.

## 8.1 Semantics

In the process of analysing the results of discourse classification, it became evident to us that some relations are semantically so similar that distinguishing between them is challenging. In these instances, even though the model may label a relation differently from the gold standard, the alternative label can still be considered correct based on the meaning that is being conveyed. This is not a matter of parser or model error. Rather, it highlights that the cues available are primarily semantic – essentially, the meaning is conveyed by the greater context within the text rather than any explicit or surface-level signals such as specific lexical items or syntactic structures. It requires an understanding of the underlying relationships and intentions between parts of the discourse based on the conceptual content rather than the observable form. In essence, the issue lies in the inherent ambiguity where the semantic information suggests multiple interpretations of the discourse relation, which can be equally valid.

Let us look at some of the misclassification examples that occur due to relation types being so semantically close that both the gold label and the label suggested by the model can be considered correct.

The examples contain information in the following format:

`connective | gold label | suggested label`

For all examples in this chapter, the translation from Czech to English was performed using machine translation, which we then manually checked and adjusted.

### Example 1:

`však | restrictive opposition | opposition`

**Context:** *Spojené státy údajně předložily Kubě nové návrhy na zvýšení roční kvóty pro legální imigranty z ostrova. Američané **však** zatím nechtějí přesná čísla zveřejnit.*

**Translation:** *The United States has reportedly presented Cuba with new proposals to increase the annual quota for legal immigrants from the island. **However**, the Americans do not wish to disclose exact numbers at this time.*

In this example, the connective *však* (translation: "however") signals a contrast between two ideas, which is why labelling it as an instance of opposition is appropriate. It clashes the expectation with the contrasting reality: despite the proposals, Americans have not disclosed precise numbers.

At the same time, this contrast is not a broad negation of the information given before. It can be interpreted as a restriction of the scope of the opposition to a particular aspect – the act of disclosing numbers. In this interpretation, *však* is signalling restrictive opposition because the negation is limited.

Therefore, both labels can be seen as valid. The label of **opposition** captures the contrast, while the label of **restrictive opposition** shows that the contrast is not of the entire context, but of a specific, limited part.

### Example 2:

však | confrontation | opposition

**Context:** *V současné době jsou povinni platit například za zboží objednané ze zahraničního katalogu buď prostřednictvím svého devizového konta, anebo cizí měnou, kterou si vymění do výše devizového limitu. Podle mluvčího České národní banky bude limit pro tyto účely s největší pravděpodobností zrušen. Vývoz hotovosti do zahraničí **však** zůstane i nadále omezen, o limitní částce se jedná.*

**Translation:** *At present, they are required to pay, for example, for goods ordered from a foreign catalogue either through their foreign exchange account or with foreign currency exchanged up to the foreign exchange limit. According to the spokesperson of the Czech National Bank, this limit will most likely be lifted for these purposes. **However**, the export of cash abroad will remain restricted, and the limit amount is still under discussion.*

In this example, the connective *však* (translation: "however") signals a contrast. On one hand, the relation can be seen as an instance of **opposition**, because the second statement simply contrasts with the first: while one limit is likely to be lifted, another restriction remains in place. This interpretation focuses on the contrast between two different regulatory outcomes without implying a direct clash.

On the other hand, the relation can also be interpreted as a case of **confrontation**. The contrast then is a direct challenge to the expectation set up by the previous context: the expectation that reforms will lead to a broader lifting of restrictions is directly countered by the maintained limitation on cash export. This interpretation points to a more active tension between policy directions – a confrontation between the easing of some controls and the persistence of others.

So both labels are appropriate: **opposition** captures the general idea of contrasting regulatory measures, and **confrontation** underscores the confrontational nature between the anticipated policy shift and the reality.

### Example 3:

nebo | conjunctive alternative | disjunctive alternative

**Context:** *Je zřetelně vidět, že ceny akcií investičních fondů si ještě nenašly svůj rovnovážný stav a jen velmi těžko tvrdit, kde by se měly pohybovat. Podle studie BH Securities jsou akcie fondů stále ještě o 20 až 80 procentů podhodnoceny, otázkou je, jak dlouho jim bude vzestup na skutečnou tržní hodnotu trvat, **nebo** zda je vůbec tak vysoká v očích českých investorů.*

**Translation:** *It is clearly evident that the share prices of investment funds have not yet found their equilibrium, and it is very difficult to say where they should be. According to a study by BH Securities, fund shares are still undervalued by 20 to 80 per cent. The question is how long it will take them to rise to their true market value, **or** whether such a high value is even perceived as realistic by Czech investors.*

In this example, the connective *nebo* (translation: "or") introduces two alternatives regarding the uncertainty about the market valuation of investment fund shares. On one hand, the use of *nebo* can be interpreted as a **disjunctive alternative**: it presents two mutually exclusive questions – either the time it takes for the shares to reach their true market value is a concern, or the perceived level of that true value is in question in the eyes of investors. This interpretation implies that only one of these possibilities is the primary issue.

On the other hand, it can also be seen as an instance of **conjunctive alternative**. In this interpretation, *nebo* does not force an either–or choice but groups the two issues as contributing to the overall uncertainty together. The discussion is not limited to a single factor. Instead, both might simultaneously play their roles in the undervaluation of shares.

Therefore, we can interpret the example as either containing two mutually exclusive options or as pointing to multiple aspects of a complex financial assessment.

## 8.2 Annotation Errors

We also observed that many of the misclassifications are not simply errors. Using the examples below, we will discuss cases where the gold standard and the model's predictions differ due to subtle nuances in meaning, which can be debated over by human annotators. We will argue that in these instances, the label suggested by the model is better than the gold label.

### Example 1:

nebo | conjunctive alternative | disjunctive alternative

**Context:** *Správa majetku zbaví klienta, který tímto majetkem oplývá, starostí se zajišťováním jeho administrativní a právní údržby či případné prosperity. Předpokládám ale, že i zde se časem projeví ocenění profesionálů v činnostech, které mají odborný charakter, že zmizí konečně éra všumělectví a že schopní lidé se dnes již věnují pouze své profesi, neboť na amatérské působení mimo svou odbornost již nemají čas **nebo** se jim to prostě nevyplácí.*

**Translation:** *Asset management relieves the client who possesses such assets of the burden of ensuring their administrative and legal maintenance or potential*

growth. However, I assume that even in this area, professional expertise in specialized activities will gradually gain recognition, that the era of jack-of-all-trades will finally come to an end, and that capable individuals nowadays focus solely on their profession, as they no longer have time for amateur involvement outside their field of expertise, **or** it simply isn't worth it for them.

In this example, there are two separate alternative reasons why capable individuals now devote themselves solely to their professions. The use of *nebo* (translation: "or") here does not imply that both reasons must be true at the same time, but that either of the two is enough to explain the phenomenon in question. In other words, it offers a choice between two independent explanations (lacking time or lacking profit), which means the relation type should be **disjunctive alternative**.

However, a **conjunctive alternative** label would suggest that both conditions have to be met at the same time to justify the claim, which is not supported by the structure or meaning of the sentence. Thus, in our opinion, the label of **disjunctive alternative** better reflects the intended meaning of the connective *nebo* in this context.

### Example 2:

navíc | conjunction | gradation

**Context:** *To je však omyl. Politické důvody tomu sice nebrání, ale náklady ano. Nevelká odezva jazzových koncertů **navíc** odrazuje i ten malý počet případných sponzorů.*

**Translation:** *However, that is a misconception. While there are no political obstacles, the costs are there. The limited response to jazz concerts **further** discourages even the small number of potential sponsors.*

In this example, the connective *navíc* (translation: "further") is used to build up the argument by adding an additional reinforcing factor, not just linking two independent clauses. The label **gradation** then captures this cumulative nature: after establishing that high costs are a barrier, the sentence then adds that the limited response at jazz concerts further discourages the few potential sponsors. This additional point does not just stand side by side with the previous one (as a simple conjunction might imply), it intensifies the overall argument.

In contrast, labelling it as **conjunction** would mean there is no sequential build-up to the conclusion. Therefore, **gradation** better reflects the intended layered contribution of reasons in this case.

### Example 3:

a to | conjunction | specification

**Context:** *Naposledy jsme se zabývali některými otázkami spojenými s výpovědí z nájmu, **a to** nikoliv z hlediska výkladu ustanovení občanského zákoníku, ale z hlediska pomalu se tvořící soudní judikatury.*

**Translation:** *Last time, we addressed some issues related to lease termination, **namely** not from the perspective of interpreting the provisions of the civil code, but from the standpoint of the gradually emerging judicial case law.*

In this example, the connective *a to* (translation: "*namely*") does not just join two independent clauses, but specifies the particular perspective on the topic. It introduces a clarification: the reader is made aware of the more specific aspect of the topic that was discussed.

Therefore, labelling this connective as **specification** more accurately captures its function, while a simple **conjunction** label would overlook this.

In conclusion, the misclassifications discussed in this chapter underline a fundamental challenge in discourse relation classification, which is the ambiguity of natural language. The examples we analysed demonstrate the complexity of choosing the appropriate label, and this complexity inevitably impacts automated systems.

Recognizing this problem is essential for further improvement of the discourse relations classification.

## 9 Discussion

The results of the experiments show that more advanced models outperform the baseline (Most Frequent Type for Each Connective), which is simple, yet still strong. The connective information by itself presented powerful performance, and the subsequent introduction of additional features improved the results slightly, especially in the case of FFNN. However, compared to the difference between the baseline and the models with just connective information, their impact was not big. This suggests that the chosen features, while useful for certain individual classes, may not be sufficiently discriminative for the overall task.

The impact of different feature sets emphasizes a critical point: the complexity of discourse relations may not be fully captured by surface-level linguistic features alone. However, further experiments and analysis reveal it not to be the only problem. The experiment on four major classes showed that while the core distinctions between discourse types are being captured successfully, the persistent challenges of class imbalance and limited representation remain, which further reinforces that class imbalance is an issue. Data augmentation to solve class imbalance might be one of the possible future directions of research.

An important factor that emerged from the analysis of misclassifications was the imperfect inter-annotator agreement in the data. If we look at the previously studied annotator agreement on the Prague Discourse Treebank 1.0[19], the accuracy of inter-annotator agreement on discourse types in PDiT 1.0 is only 0.77. Comparably, for English data in PDTB 2.0[20], the inter-annotator agreement is 0.8. The annotations for the updated versions of PDiT were revised several times, meaning now the quality is improved compared to the older study, but it has not been measured this way since then. Still, it is unlikely that the issue of inconsistency in the inter-annotator agreement has been resolved to perfection, and the challenge remains. For some cases, we uncovered that the labels proposed by the models seemed more reasonable than the gold standard annotations, suggesting that human annotators' subjective interpretations may contribute to misclassifications.

We think it is interesting to mention a previously conducted study on the classification of discourse relation[21]. It shows similar trends to our experiments: strong performance of a similarly defined baseline (Most Frequent Type for Each Connective) and a modest improvement in accuracy by advanced BERT-based models, achieving the best accuracy of 0.79 for Czech data. However, it cannot be directly compared to our results due to the fact that the labels for classification were defined differently, and the experiments in [21] were conducted on an older version of the data.

Our additional experiment that utilises BERT-based RoboCzech showcases that with the use of greater context and a more sophisticated model, the overall results do see a boost, but individual classes on which other models struggled remain mostly unsolved, further highlighting the limitations of the dataset and inherent similarities between classes.

# Conclusion

In this study, we first explained both PDiT-PDTB and UD formats and discussed an already existing approach to discourse annotation in UD. We then proposed a new approach of integrating discourse relation information into UD, and processed the data from an already existing PDiT-PDTB together with UD annotations created by us using UDPipe to produce the dataset in the proposed format.

After that, we introduced two baselines and conducted a set of experiments on the development set of the produced dataset using different models and different sets of features, all of which outperformed the baselines. The best model, which we have chosen based on the results of evaluation on the development set, showed strong performance on the evaluation set, achieving an accuracy of 0.86. This was the FFNN model with the following set of features: connective form and length, argument lengths, argument roots' lemmas, polarity, modal verbs counts.

The experiments showed that using just the connective information yields strong results, and while sometimes additional features may improve the performance (like in the case of our best model), other times they do not contribute to or even worsen the performance.

We also performed additional experiments, which pointed to the dataset limitations, such as Four Major Classes classification struggling with an under-represented class or a Transformer-based model not being able to gain a major advantage against simpler models on certain individual classes despite the richer context it was able to utilise. The Transformer-based model, however, showed an improvement from all other presented models and approaches, and represents a promising direction of future research.

Afterwards, we performed a misclassification analysis that revealed an additional layer of the task's complexity in the inconsistency of inter-annotator agreement.

In conclusion, the experiments in this thesis provide valuable insights into the strengths and limitations of different discourse relations classification methods, and they also reveal the complexity of the task. Despite the use of different architectures, features and incorporation of RobeCzech in the additional experiment, the fundamental issues of semantic similarity between discourse types, dataset imbalance, and human annotation inconsistencies remain prevalent, creating problems for automated classification, especially on rare and ambiguous classes.

**Future work** In future research, we see a promising direction in exploring the use of pre-trained embeddings (GloVe[22], fastText[23]) and Large Language Models (LLMs). Pre-trained embeddings could provide additional context and nuanced semantic information that might not be captured by the current methods. By fine-tuning these embeddings, it may be possible to enhance a model's understanding of subtle distinctions between discourse types using string-based features. Additionally, for work with more focus on improving the automated discourse classification rather than feature contribution, leveraging LLMs, particularly those designed for discourse understanding (e.g., GPT-like models[24] or BERT-based

models[25]), could significantly boost performance. LLMs excel at capturing complex long-range dependencies and contextual relationships, which could help overcome some of the challenges seen in this study.

# Bibliography

1. SYNKOVÁ, Pavlína; MÍROVSKÝ, Jiří; POLÁKOVÁ, Lucie; RYSOVÁ, Magdaléna. Announcing the Prague Discourse Treebank 3.0. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 1270–1279.
2. HAJIČ, Jan; BEJČEK, Eduard; BÉMOVÁ, Alevtina; BURÁŇOVÁ, Eva; FUČÍKOVÁ, Eva; HAJIČOVÁ, Eva; HAVELKA, Jiří; HLAVÁČOVÁ, Jaroslava; HOMOLA, Petr; IRCING, Pavel; KÁRNÍK, Jiří; KETTNEROVÁ, Václava; KLYUEVA, Natalia; KOLÁŘOVÁ, Veronika; KUČOVÁ, Lucie; LOPATKOVÁ, Markéta; MAREČEK, David; MIKULOVÁ, Marie; MÍROVSKÝ, Jiří; NEDOLUZHKO, Anna; NOVÁK, Michal; PAJAS, Petr; PANEVOVÁ, Jarmila; PETEREK, Nino; POLÁKOVÁ, Lucie; POPEL, Martin; POPELKA, Jan; ROMPOLTL, Jan; RYSOVÁ, Magdaléna; SEMECKÝ, Jiří; SGALL, Petr; SPOUSTOVÁ, Johanka; STRAKA, Milan; STRAŇÁK, Pavel; SYNKOVÁ, Pavlína; ŠEVČÍKOVÁ, Magda; ŠINDLEROVÁ, Jana; ŠTĚPÁNEK, Jan; ŠTĚPÁNKOVÁ, Barbora; TOMAN, Josef; UREŠOVÁ, Zdeňka; VIDOVÁ HLADKÁ, Barbora; ZEMAN, Daniel; ZIKÁNOVÁ, Šárka; ŽABOKRTSKÝ, Zdeněk. *Prague Dependency Treebank - Consolidated 2.0 (PDT-C 2.0)*. 2024. Available also from: <http://hdl.handle.net/11234/1-5813>. LINDAT/CLARIAH-CZ digital library, Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
3. PRASAD, Rashmi; WEBBER, Bonnie; LEE, Alan; JOSHI, Aravind. *Penn Discourse Treebank Version 3.0* [LDC Catalog No. LDC2019T05]. Philadelphia: Linguistic Data Consortium, 2019. ISBN 1-58563-877-3. Available from DOI: 10.35111/qebf-gk47. Release Date: March 15, 2019.
4. MARNEFFE, Marie-Catherine de; DOZAT, Timothy; SILVEIRA, Natalia; HAVERINEN, Katri; GINTER, Filip; NIVRE, Joakim; MANNING, Christopher D. Universal Dependencies: A Cross-Linguistic Typology. [N.d.].
5. CONTRIBUTORS, Universal Dependencies. *Universal Dependencies*. 2024. Available also from: <https://universaldependencies.org>.
6. STRAKA, Milan. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 197–207. Available from DOI: 10.18653/v1/K18-2020.
7. ZELDES, Amir. The GUM Corpus: Creating Multilayer Resources in the Classroom. *Language Resources and Evaluation*. 2017, vol. 51, no. 3, pp. 581–612. Available from DOI: <http://dx.doi.org/10.1007/s10579-016-9343-x>.
8. LI, J.; LIU, M.; QIN, B., et al. A Survey of Discourse Parsing. *Front. Comput. Sci.* 2022, vol. 16, p. 165329. Available from DOI: 10.1007/s11704-021-0500-z.

9. MÍROVSKÝ, Jiří; RYSOVÁ, Magdaléna; SYNKOVÁ, Pavlína; POLÁKOVÁ, Lucie. Prague to Penn Discourse Transformation. *The Prague Bulletin of Mathematical Linguistics*. 2023, no. 120.
10. SCIKIT-LEARN DEVELOPERS. *sklearn.metrics*. scikit-learn, 2025. Version 1.6.1. Available also from: <https://scikit-learn.org/stable/api/sklearn.metrics.html>.
11. BREIMAN, Leo. Random forests. *Machine learning*. 2001, vol. 45, pp. 5–32.
12. GOYAL, Palash; PANDEY, Sumit; JAIN, Karan. Deep Learning for Natural Language Processing. *New York: Apress*. 2018.
13. LECUN, Yann; BENGIO, Yoshua, et al. Convolutional Networks for Images, Speech, and Time Series. *The handbook of brain theory and neural networks*. 1995, vol. 3361, no. 10, p. 1995.
14. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. *Neural computation*. 1997, vol. 9, no. 8, pp. 1735–1780.
15. PYTORCH. *torch.nn.Embedding*. 2025. Available also from: <https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html>.
16. HERINEK, Pavel. The Comparison of Modal Verbs in English, Spanish and Czech. [N.d.].
17. JÍNOVÁ, Pavlína; POLÁKOVÁ, Lucie; MÍROVSKÝ, Jiří. Sentence Structure and Discourse Structure: Possible Parallels. In: *Dependency Linguistics*. John Benjamins Publishing Company, 2014, pp. 53–74.
18. STRAKA, Milan; NÁPLAVA, Jakub; STRAKOVÁ, Jana; SAMUEL, David. RobeCzech: Czech RoBERTa, a Monolingual Contextualized Language Representation Model. In: EKŠTEIN, Kamil; PÁRTL, František; KONOPÍK, Miloslav (eds.). *Text, Speech, and Dialogue. TSD 2021*. Springer, Cham, 2021, vol. 12848, pp. 197–209. Lecture Notes in Computer Science. Available from DOI: 10.1007/978-3-030-83527-9\_17.
19. POLÁKOVÁ, Lucie; MÍROVSKÝ, Jiří; NEDOLUZHKO, Anna; JÍNOVÁ, Pavlína; ZIKÁNOVÁ, Šárka; HAJICOVÁ, Eva. Introducing the Prague Discourse Treebank 1.0. In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 2013, pp. 91–99.
20. PRASAD, N. Dinesh; LEE, A.; AL., et. *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*. The Penn Discourse Treebank 2.0. Marrakech, Morocco, 2008.
21. MÍROVSKÝ, Jiří; POLÁKOVÁ, Lucie. Sense Prediction for Explicit Discourse Relations with BERT. In: *Proceedings of Sixth International Congress on Information and Communication Technology: ICICT 2021, London, Volume 3*. Springer, 2022, pp. 835–842.
22. PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D. Glove: Global Vectors for Word Representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

23. JOULIN, Armand; GRAVE, Edouard; BOJANOWSKI, Piotr; MIKOLOV, Tomas. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759* 2016.
24. BROWN, Tom; MANN, Benjamin; RYDER, Nick; SUBBIAH, Melanie; KAPLAN, Jared D; DHARIWAL, Prafulla; NEELAKANTAN, Arvind; SHYAM, Pranav; SASTRY, Girish; ASKELL, Amanda, et al. Language Models are Few-Shot Learners. *Advances in neural information processing systems*. 2020, vol. 33, pp. 1877–1901.
25. DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186.

# List of Tables

4.1	Global Most Frequent Type Classification Report on Evaluation Directory . . . . .	21
4.2	Most Frequent Type for Connective Classification Report on Evaluation Directory . . . . .	22
6.1	FFNN Classification Report – Experiment 1.1 . . . . .	29
6.2	FFNN Classification Report – Experiment 1.2 . . . . .	30
6.3	FFNN Classification Report – Experiment 1.3 . . . . .	30
6.4	FFNN Classification Report – Experiment 1.4 . . . . .	30
6.5	FFNN Classification Report – Experiment 1.5 . . . . .	31
6.6	FFNN Classification Report – Experiment 1.6 . . . . .	31
6.7	Best FFNN Classification Report on Evaluation Set . . . . .	32
6.8	RNN Classification Report – Experiment 2.1 . . . . .	33
6.9	RNN Classification Report – Experiment 2.2 . . . . .	34
6.10	RNN Classification Report – Experiment 2.3 . . . . .	34
6.11	RNN Classification Report – Experiment 2.4 . . . . .	35
6.12	RNN Classification Report – Experiment 2.5 . . . . .	35
6.13	RNN Classification Report – Experiment 2.6 . . . . .	35
6.14	Best RNN Classification Report on Evaluation Set . . . . .	36
6.15	CNN Classification Report – Experiment 3.1 . . . . .	37
6.16	CNN Classification Report – Experiment 3.2 . . . . .	37
6.17	CNN Classification Report – Experiment 3.3 . . . . .	38
6.18	CNN Classification Report – Experiment 3.4 . . . . .	38
6.19	CNN Classification Report – Experiment 3.5 . . . . .	38
6.20	CNN Classification Report – Experiment 3.6 . . . . .	39
6.21	Best CNN Classification Report on Evaluation Set . . . . .	39
7.1	Classification Report for Major Discourse Classes – Baseline . . . . .	41
7.2	Classification Report for Four Major Classes – Best Model . . . . .	42
7.3	FFNN and Baseline Ensemble Classification Report on Evaluation Set . . . . .	43
7.4	Voting Classification Report on Evaluation Set . . . . .	44
7.5	Comparison to the Best Models . . . . .	45
7.6	Transformer Classification Report on Evaluation Set . . . . .	45
A.1	List of all class senses and with their global and relative frequencies.	60
A.2	List of original discourse types in Prague taxonomy with their global and relative frequencies. . . . .	61
A.3	Grouping of classes into four major classes. . . . .	61
A.4	Top 10 misclassified connectives . . . . .	62

# List of Abbreviations

**PDiT** – Prague Discourse Treebank

**PDTB** – Penn Discourse Treebank

**PDiT-PDTB** – PDiT represented in PDTB format

**UD** – Universal Dependencies

**GUM** – Georgetown University Multilayer corpus

**FFNN** – Feed-Forward Neural Network

**CNN** – Convolutional Neural Network

**RNN** – Recurrent Neural Network

**LSTM** – Long Short-Term Memory

**LLM** – Large Language Model

**ReLU** – Rectified Linear Unit

# A Appendices

## A.1 List of Class Senses

Discourse Relation	Global	Train	Dev	Eval
Expansion.Conjunction	8160	6323	877	960
Comparison.Concession.Arg2-as-denier	3551	2772	377	402
Contingency.Cause.Reason	1749	1352	192	205
Contingency.Cause.Result	1298	1007	146	145
Contingency.Condition.Arg2-as-cond	1237	960	128	149
Comparison.Contrast	780	626	67	87
Temporal.Asynchronous.Precedence	686	506	82	98
Expansion.Level-of-detail.Arg2-as-detail	644	505	59	80
Comparison.Concession.Arg1-as-denier	568	454	55	59
Contingency.Purpose.Arg2-as-goal	415	329	45	41
Expansion.Substitution.Arg2-as-subst	390	313	45	32
Expansion.Disjunction	367	277	51	39
Temporal.Asynchronous.Succession	341	258	36	47
Temporal.Synchronous	262	191	25	46
Expansion.Instantiation.Arg2-as-instance	205	165	22	18
Expansion.Exception.Arg2-as-excpt	195	154	21	20
Expansion.Level-of-detail.Arg1-as-detail	136	106	14	16
Expansion.Equivalence	127	103	13	11
Contingency.Cause+Belief.Reason+Belief	122	99	14	9
Contingency.Condition+SpeechAct	102	70	13	19
Expansion.Substitution.Arg1-as-subst	60	46	8	6
Contingency.Condition.Arg1-as-cond	48	34	5	9
Contingency.Negative-condition.Arg2-as-negCond	48	38	3	7
Comparison.Similarity	47	38	1	8
Contingency.Negative-cause.NegResult	8	8	0	0
Contingency.Cause+Belief.Result+Belief	7	7	0	0
Expansion.Exception.Arg1-as-excpt	6	5	1	0
Contingency.Purpose.Arg1-as-goal	6	6	0	0
Contingency.Cause+SpeechAct.Result+SpeechAct	4	4	0	0
Comparison.Concession+SpeechAct.Arg2-as-denier+SpeechAct	4	4	0	0
Expansion.Instantiation.Arg1-as-instance	2	2	0	0
Contingency.Negative-condition.Arg1-as-negCond	2	0	2	0
Contingency.Cause+SpeechAct.Reason+SpeechAct	2	1	1	0

**Table A.1** List of all class senses and with their global and relative frequencies.

## A.2 Original Discourse Types in Prague Taxonomy

Discourse Relation	Global	Train	Dev	Eval
conjunction	7739	6019	816	904
opposition	3202	2493	343	366
reason-result	3023	2339	335	349
condition	1331	1028	138	165
precedence-succession	1027	764	118	145
concession	901	722	87	92
confrontation	686	556	57	73
specification	609	476	57	76
gradation	468	342	62	64
correction	450	359	53	38
purpose	421	335	45	41
restrictive opposition	285	220	31	34
disjunctive alternative	271	211	34	26
synchrony	262	191	25	46
instantiation	207	167	22	18
explication	146	119	16	11
generalization	136	107	14	15
equivalence	127	103	13	11
pragmatic condition	106	73	13	20
conjunctive alternative	96	66	17	13
pragmatic reason-result	57	49	4	4
pragmatic contrast	29	24	3	2

**Table A.2** List of original discourse types in Prague taxonomy with their global and relative frequencies.

## A.3 Four Major Classes

Major Class	Sub-Classes
Temporal	synchrony, precedence-succession
Contingency	reason-result, pragmatic reason-result, condition, pragmatic condition, explication, purpose
Contrast	confrontation, opposition, pragmatic contrast, restrictive opposition, concession, correction, gradation
Expansion	conjunction, specification, equivalence, generalization, conjunctive alternative, disjunctive alternative, instantiation

**Table A.3** Grouping of classes into four major classes.

## A.4 Top 10 Misclassified Connectives

Connective	Gold Label	Suggested Label
však	restrictive opposition	opposition
nebo	conjunctive alternative	disjunctive alternative
však	confrontation	opposition
a to	conjunction	specification
navíc	conjunction	gradation
zároveň	synchrony	conjunction
pak	conjunction	precedence-succession
ale	restrictive opposition	opposition
navíc	gradation	conjunction
když	specification	condition

Table A.4 Top 10 misclassified connectives

## A.5 User Guide

**Overview** The project allows the user to:

1. Create UD annotations using UDPipe.
2. Align UD and PDiT-PDTB annotations into a new combined format proposed in the study.
3. Explore different sets of features with FFNN / CNN / RNN classifiers, both to train models and evaluate the models discussed in the study.
4. Run additional experiments (Four Major Classes, Baseline+Model Ensemble, Voting, Transformer-based Model using RoboCzech).

### Requirements

- This project requires Python 3 and was developed and tested on **macOS** (a Unix-based system) using **Python 3.9.6**
- 
- RAM: **16 GB**
- Free disk space: **64 GB**

To install the requirements, we recommend to first create a virtual environment, then run:

```
pip install -r requirements.txt
```

**Available data and models** The project contains the collection of raw texts, PDiT-PDTB dataset, created UD annotations using UDPipe, annotations in the new format described in the study, trained models used for evaluation on development and evaluation sets and built vocabularies.

**How to run** All entry-points live in `code/__init__.py`. In its `main()` the user will find commented calls for each experiment. All function calls are labelled in `code/__init__.py`. To run the needed experiment, uncomment the appropriate function call, choose the desired set for evaluation (both `dev_set` and `eval_set` are defined, by default `dev_set` is chosen) and run from `code/:` `PYTHONPATH=".." python -m code`.

**Examples** The following are a few examples on how to run some of the experiments. More of the specifics can be found in `USER.md` and `DEVELOPER.md` documentations in the project.

**To run baselines:**

1. Uncomment `run_baselines(base_path, train_dirs, dev_dirs)`
2. Run from `code/:` `PYTHONPATH=".." python -m code`

**To run FFNN experiments for evaluation of the trained models:**

1. Uncomment `run_ffnn_experiments(base_path, train_dirs, dev_dirs, mode="test")`
2. Run from `code/:` `PYTHONPATH=".." python -m code`

**To run Voting experiment:**

1. Uncomment `run_voting(base_path, train_dirs, dev_dirs)`
2. Run from `code/:` `PYTHONPATH=".." python -m code`

Note that when rerunning the evaluation, there might be some numerical discrepancies in due to differences in hardware, so the numbers may differ slightly.